

ファミリーベーシックV3™
FAMILY BASIC V3
ハンドブック



Nintendo®

ファミリーベーシック

FAMILY BASIC™ V3

ハンドブック

スクリプトの構成要素	1
3Dグラフィックだけをメモリバックアップする場合	20
メモリバックアップしたものを使える状態に戻す手順	21
その他の初期化	22
3DグラフィックをもつてJ画面も表示できるのか	23
アーケード機でもあり本格的な筐体でも動作するか	24
Jより本格的な容量リセット	25
家庭用と日本用V3の違いについて	26
日本用ソフトウェアの改造で命が危ない	27
メモリの初期化	28



ごあいさつ

このたびは、任天堂『ファミリーベーシックV3』カセットをお買いあげいただきまして、誠にありがとうございました。ご使用の前に、取り扱い方、使用上の注意等、このハンドブックをよくお読みいただき、正しい使用方法でご愛用ください。なお、このハンドブックは、大切に保管してください。

もくじ

ファミリー ベーシック	1 FAMILY BASIC V3の特徴を知ろう！	1
1-1	4つのプログラムを内蔵しています	2
1-2	BGグラフィックが使いやすくなりました	4
1-3	メモリ容量が2倍になりました	6
1-4	BASICが強力になりました	7
A.	プログラム作りを助ける命令	8
B.	ゲーム作りに役立つ命令	9
C.	BGグラフィックの命令	10
D.	その他の命令	11
ファミリー ベーシック	2 FAMILY BASIC V3の基本操作	13
2-1	接続のしかた	14
2-2	電源の入れかた	15
①	キーボードのコネクタを差し込む	15
②	V3カセットを挿入する	15
③	電源をONにする	15
④	BASIC準備よし！	15

2-3 内蔵プログラムの使い方

Ⓐ ファンクションキーで呼び出す方法 17
Ⓑ コマンドを入力して呼び出す方法 17

2-4 メモリバックアップについて

A. プログラムとBGグラフィックの両方をメモリバックアップする場合 19
B. プログラムだけをメモリバックアップする場合 20
C. BGグラフィックだけをメモリバックアップする場合 20
D. メモリバックアップしたものを使える状態に戻す手順 21

2-5 カセットテープへの保存と呼び出し

● 内蔵プログラムについて 23
● プログラムとBGグラフィックと一緒にSAVE/LOADする方法 24

3 FAMILY BASIC V3のゲームプログラム

3-1 プログラムの改造方法

① 文字変更したいとき 26
② 間に何文字かを入れたいとき 27
③ 何文字かを詰めたいとき 27
④ 新しい行を追加したいとき 27
⑤ 1行を全部変更したいとき 28
⑥ 1行を全部消したいとき 28
⑦ 行と行がつながってはいけません 28
⑧ 内蔵プログラムを改造する前に 29

3-2 内蔵プログラムで遊ぼう

1. ハート 30
A. 遊び方 30

B. 改造の方法	31
①ハートの色を変えてみましょう	31
②バックの色を変えてみましょう	33
③ハートを描くキャラクタを変えてみましょう	34
④タイマーを付けてみましょう	35
C. マイクを使ってみましょう	37
①マイクの使い方	37
②肺活量遊び	38
ハート・プログラムリスト	40
2. ペンペン迷路	43
A. 遊び方	43
B. 改造の方法	44
①色を変えてみましょう	45
②ペンペンやカニさんをほかのアニメキャラクタに変えてみましょう	45
③得点を変えてみましょう	46
④ワナの数を変えてみましょう	46
⑤カニさんの数を変えてみましょう	47
⑥ワナをうまくおくことができないときは	47
⑦ペンペンが拾えるものを数字以外のキャラクタにしてみましょう	48
⑧プレイ中に次に何を拾うかを画面に出してみましょう	49
⑨ボーナス得点をつけましょう	49
⑩ボーナスキラクタを拾うとカニさんが逃げるようにしましょう	50
⑪タイマーを作つてみましょう	51
⑫カニさんの動きを変えてみましょう	53
⑬カニさんが追いかけてくる速さを変えてみましょう	54

ペンペン迷路・プログラムリスト	56
3. マリオワールド	61
A. 遊び方	61
B. 改造の方法	62
①BGグラフィックを変えてみましょう	62
②BGグラフィックの色を変えてみましょう	63
③アニメキャラクタを変えてみましょう	64
④二タニタの数を変えてみましょう	65
⑤得点を変えてみましょう	65
⑥マリオが最初に出てくる場所を変えてみましょう	66
⑦タイマーを付けてゲーム時間を変えてみましょう	66
⑧ジャンプ台でのジャンプの高さを変えてみましょう	67
⑨Aボタンでのジャンプ幅を変えてみましょう	68
⑩Aボタンでのジャンプ力を変えてみましょう	69
⑪マリオの歩き始めをすばやくしましょう	71
マリオワールド・プログラムリスト	72
マリオワールド・BGグラフィック	77
4. スターキラー	78
A. 遊び方	78
B. 改造の方法	80
①最初に持っている宇宙船の数を変えてみましょう	80
②出てくる飛行体の数を変えてみましょう	80
③宇宙船のスピードを変えてみましょう	81
④1人用、2人用を選ぶ画面をなくしてみましょう	82
⑤邪魔ものをあいてみましょう	84

◎基地を作つてみましよう.....	85
スタークリー・プログラムリスト.....	87
DEF MOVEとDEF SPRITE.....	92

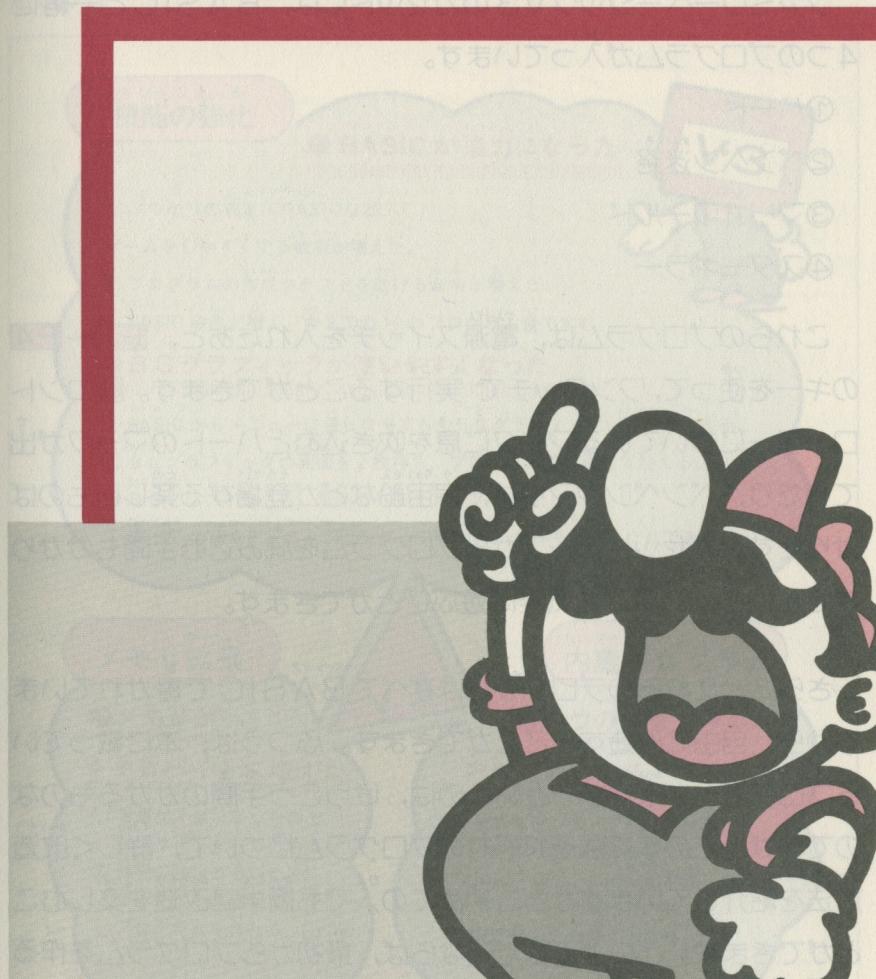
4 FAMILY BASIC V3の命令早わかり..... 95

本文の読み方.....	96
AUTO 97 / BACKUP 98 / BGGET 99 / BGPUT 99 / BGTOOL 100 / CAN 101 / CLICK 102 / CLS 102 / CRAS DELETE 103 / ERL 104 / ERR 105 / ERROR 106 FIND 106 / FILTER 107 / GAME 108 / INSTR 109 / LOAD S 110 / ON ERROR GOTO 111 / RENUM 112 / RESUME SAVES 113 / SCREEN 114 / TRON / TROFF 116 / VCT 117	※これらは、V3カセットで新しく追加された命令です。

5 ふろく 119

BGTOOL でのキーの働き スタート (電源ONから) の手順 内蔵プログラムの実行と変更 終了 (電源OFF) の手順 メモリマップ エラーメッセージ スプライトの関数値とアニメキャラクタの状態 DEF MOVE系とDEF SPRITE系の命令対照表 BGキャラクタをリストに使う方法 キャラクタテーブル キャラクタコード表 ファミリーベーシックV3の仕様 BASICの仕様水準 取り扱い上のご注意 / 乾電池取り扱い上のご注意 ご注意 アルファベット順命令語索引	120 121 122 125 126 126 128 129 130 131 132 136 137 138 139 141
---	--

ファミリーベーシック 1. FAMILY BASIC™ V3 の 特徴を知ろう!



1-1

4つのプログラムを 内蔵しています

ファミリーベーシックV3のカセットには、BASICと一緒に
4つのプログラムが入っています。

- ①ハート
- ②ペンペン迷路
- ③マリオワールド
- ④スタークィラー

これらのプログラムは、電源スイッチを入れたあと、**F1**～**F4**のキーを使って、ワンタッチで実行することができます。IIコントローラーについているマイクに息を吹き込むとハートのマークが出てきたり、ペンペンやマリオ、宇宙船などが登場する楽しいものばかりです。カセットテープからプログラムを読み込む手間もかかりませんから、いつでも気軽に遊ぶことができます。

さらに、これらのプログラムはすべてBASICで書かれていますから、自由に改造することができます。ふつうは、本に載っているプログラムを改造するというのは、けっこう手間のかかるもののですが、この本ではそれぞれのプログラムについて、詳しく改造方法を紹介していますから、初めての人でも簡単に改造を楽しむことができます。このような方法ならば、最初からプログラムを作ることなく、自分だけのオリジナルゲームを作ることができるわけです。

なあ、内蔵されている4つのプログラムについては、全部のリストとその説明がしてあります。細かい点まで理解するのは難しいかも知れませんが、プログラムの全体的な流れはつかめると思います。ぜひ、BASICの勉強や自分でゲームを作るときの参考にしてください。

機能の強化

● **BASIC** が強力になった

- ① ぶつかりの判定(CRASH)など、ゲームをしやすくする機能が増えた。
- ② プログラムの作成やテストを助ける命令が増えた。
- ③ BASIC 命令が増え、今までの約3倍のプログラム量ですむ。

● **BG グラフィック** が使いやすくなった

- ① BASIC からメニューに戻らなくても、BG グラフィック(作成)が使える。
- ② BG グラフィックの画面を2枚持ち、プログラムで切換えて使える。
- ③ BG グラフィックも、電池でバックアップできる。
- ④ BASIC の命令で、BG 画面をテープに保存したり呼び出したりできる。

メモリ拡張

● **メモリ** が4キロバイトに増えた

- ① 容量が2倍になったので、長いリストを打ち込める。
- ② 機能の強化と合わせて、今までより高度なプログラムを作れる。

内蔵プログラム

● 4つのプログラムが内蔵されている

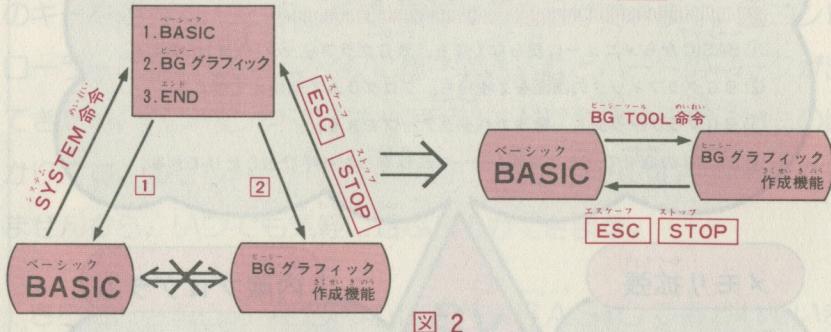
- ① そのまま使って、すぐに遊べる。
- ② いろいろな改造方法を紹介してあるので簡単にプログラムの改造が楽しめる。
- ③ リストに沿った解説付きなので、プログラムの勉強にもなる。

1-2

BGグラフィックが使いやすくなりました

今までのファミリーベーシックでは、BASICのプログラムを作る機能と、背景画(BGグラフィック)を作る機能を、メニュー画面で選ぶようになっていました。このため、プログラムを作っている途中で、BGグラフィックを変更したくなつた場合など、一度メニュー画面に戻つて、あらためてBGグラフィックを選択しなおさなくてはなりませんでした。

これまでのファミリーベーシック

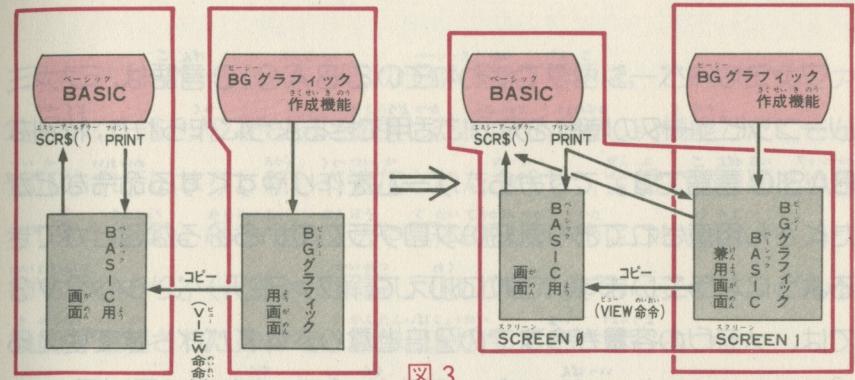


しかし、ファミリーベーシックV3では、BGグラフィックを作成する機能に、「BG TOOL」という名前をつけ、BASICから命令で直接呼び出せるようになっています。また、BGグラフィックの画面をカセットテープに保存(セーブ)したり呼び出し(ロード)たりする場合も、BASICの命令ひとつで直接実行できるようになりました。さらにプログラムを電池でバックアップするときに、

BGグラフィックも一緒にバックアップできるようになりましたから、作りかけのゲームを保存しておきたいときなどとても便利です。

これまでのファミリーベーシック

V3



BGグラフィック関係でもうひとつ強化された点は、BGグラフィックが2面使えるようになったことです。今までのファミリーベーシックでも、「BGグラフィックを作るときの画面」と「BASICで使う画面」という2つの画面がありましたが、実際にプログラムで使えるのはBASIC用の画面だけで、ここにBGグラフィック用の画面をコピー(VIEW命令)して使うようになりました。つまり、プログラム実行中は、BGグラフィック作成用の画面は使われていなかつたわけです。

ところが、ファミリーベーシックV3では、両方の画面をプログラムで使うことができます。つまり、BGグラフィック作成用の画面に對しても、PRINT命令で文字や記号を書いたり、SCR\$で画面の文字や記号を読み取つたりできるというわけです。もちろん、BGグラフィック作成用の画面を、もう一方の画面にコピーする機能も残っていますから、今までの使い方と併せていろいろと応用できるでしょう。

1-3

メモリ容量が 2倍になりました

ファミリーベーシックで使われているBASIC言語は、ファミリーコンピュータの特徴を充分に活用できるように作られた、特別なBASIC言語です。ですから、ゲームを作りやすくする命令などが多く用意されており、短いプログラムでいろいろなことができるようになっています。これに加えて、ファミリーベーシックV3では、メモリの容量が今までの2倍となり、4Kバイトまで使えるようになりました。一般的のパソコンと比べて、少ないメモリでも、独特的な特徴ある命令を活用することによって、短くて効率のよいプログラムを作ることができます。



1-4

BASICが 強力になりました

ファミリーベーシック V3のBASIC言語は、これまでのファミリーベーシックをさらに拡張したもので、新しい機能がたくさん追加されています。新しく追加された命令は、全部で26種類あります(図4参照)。各命令の文法や具体的な使い方については、「4. 命令早わかり」(P.95)で解説してありますので、ここでは、どんな機能が追加されたのかを、簡単に紹介しておきましょう。

●プログラム作りを助ける命令 (11種)

リナンバー オート デリート
RENUM・AUTO・DELETE
トレースオン トレースオフ ファインド
TRON・TROFF・FIND
オン エラー ゴーツー リジューム
ON ERROR GOTO・RESUME
イーアールエル イーアールアール エラー
ERL・ERR・ERROR

●ゲーム作りに役立つ命令 (6種)

キャンセル クラッシュ ベクター
CAN・CRASH()・VCT()
スクリーン シーエルエス
SCREEN・CLS
フィルター
FILTER

●BGグラフィックの命令 (5種)

ビージーツール
BGTOOL・
ビージーフット ビージーフット
BGET・BPUT
セーブ エス ロード エス
SAVES・LOADS

●その他の命令 (4種)

バックアップ クリック オン オフ
BACKUP・CLICK ON/OFF
インストリング ゲーム
INSTR・GAME

A. プログラム作りを助ける命令

まずは、プログラムを新しく打ち込んだり、打ち込んだプログラムのテストをしたりするときに便利な命令を紹介しましょう。たとえば、プログラムを打ち込むときに、行番号を自動的につけてくれるAUTO命令や行を追加・削除して、行番号がふぞろいになった場合にそろえ直すRENUM命令など、使い慣れるとなかなか便利なものです。また、プログラムの動きをテストするTRON/TROFF命令や、リストの中から必要な命令や変数名を探し出すFIND命令なども追加され、さらに使いやすくなりました。

●RENUM

すでに打ち込んであるリストの行番号を、10, 20………というように、きれいにそろえ直す命令。

●AUTO

プログラムを打ち込むときに、行番号を自動的に付けてくれる命令。

●DELETE

すでに打ち込んであるリストの、何行分かをまとめて消す命令。

●TRON/TROFF

プログラムを実行させたとき、リストの行番号を実行順に表示させる命令と、実行を元の状態に戻す命令。

●FIND

指定した文字列をリスト中から探し、該当する行を表示させる命令。「〇〇」という命令を使っている行をリストアップする」といつ



つか
た使い方でできます。

●ON ERROR GOTO/RESUME

プログラムを実行させたとき、「エラーが起きたら〇〇行にジャンプする」という指定をしておく命令。そして、ジャンプした後でもう1度元の行に戻すために、RESUME命令を使います。

●ERL/ERR/ERROR

プログラムの中でエラー処理をするための命令。どの行でどんなエラーが起きたかを調べたりするときに使います。

B. ゲーム作りに役立つ命令

次に、ゲームのプログラムを作るときに活用すると、効率のよくなる命令を紹介しましょう。とくに CAN, CRASH(), VCT()などは動きが複雑で、しかも速いプログラムを作りたいという場合に、大いに力を発揮してくれます。こうした命令をたくさん持っていることが、ファミリーベーシックV3が短くて効率のよいプログラムを作れる秘密なのです。

●CAN

DEF MOVEによる指定を取り消す命令。

●CRASH()

DEF MOVEで動かしたマリオなどのアニメキャラクタが、他のアニメキャラクタとぶつかったかどうかを調べる命令。ユーザーの命中判定などに使います。



●VCT()

MOVE 命令で移動中のアニメキャラクタの移動方向を調べる命令。

●SCREEN

BG面0とBG面1を切り替えて使うための命令。片方を表示しながら、もう一方の画面に書き込むといったこともできます。

●CLS

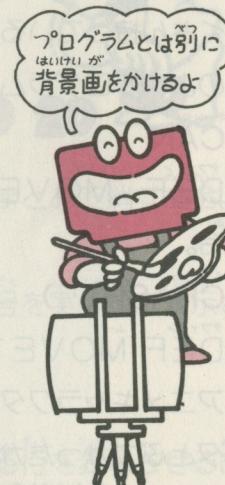
今までのCLS命令を拡張し、2つの画面のうちどちらか指定した方だけを消すことができる命令。

●FILTER

画面全体を淡く着色する命令。

C. BGグラフィックの命令

BGグラフィックは、ファミリーベーシックの大きな特徴のひとつで、プログラムとは別に背景画を描けることにより、メモリを大幅に節約できるようになっています。ファミリーベーシックV3では、さらにこの機能を使いややすくするために、BGグラフィックの画面を電池でバックアップしたり、BASICからの命令で、カセットテープに保存したり呼び出したりすることができるようになりました。ただし、これらの命令でバックアップや保存ができるのは、BG面1 (BGグラフ



イソツク作成用の画面)だけですから注意してください。

●BGTOOL

BASICから、BGグラフィックの作成機能を呼び出す命令。

●BGGET/BGPUT

BG画面を、電池でバックアップするための準備をする命令と、バックアップしたBG画面を元に戻す命令。これらの命令を忘れるとBG画面のバックアップができません。

●SAVES/LOADS

BG画面をカセットテープに保存したり、呼び出したりする命令。プログラムの保存・呼び出しは従来のSAVE/LOADですから、この命令はこれにSCREENのSをつけてあるわけです。なお、プログラムとBG画面を同時にSAVE/LOADする方法もあります (P.24参照)。

D. その他の命令

●BACKUP

プログラムやBG画面を電池でバックアップするために、実行する命令。この命令を忘ると、正しくバックアップできません。

●CLICK ON/OFF

キーを押したときに、「ピッピッ」という音 (クリック音) を出す、出さない、という指定をする命令。

●INSTR()

データ文字列の中に、指定した文字列があるかどうかを調べる命令。

●GAME

内蔵されている4つのプログラムを呼び出すための命令。呼び出すと、すぐに実行できます。プログラムの番号は0~3で、一緒に内蔵されている背景画面の呼び出しも指定できます。

*キーボードの使い方、命令、キャラクターテープルA・B、カラーチャート等については、これまでのファミリーベーシック「取扱説明書」をご覧ください。

C. BGグラフィックの命令



自分で音楽を作成したり、音楽を再生したりする機能をもつた機器の中でも、BG音楽機能をもつた機器はBG音楽機能をもつた機器です。

ファミリーベーシック

2. FAMILY BASIC

V3の
きほんそくさ
基本操作

2-1

接続のしかた

基本的な接続方法は、これまでのファミリーベーシックと変わりありません。

ただし、ファミリーコンピュータ本体に差し込むカセットが、これまでのものからV3カセット(赤)に替わります。

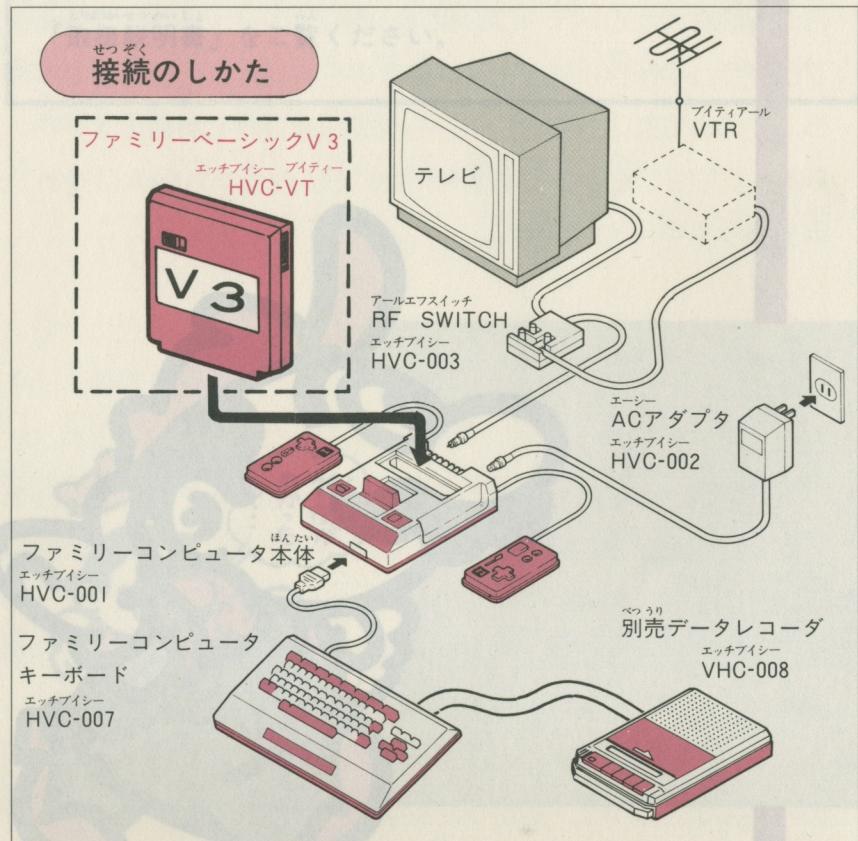


図5

2-2

電源の入れかた

①キーボードのコネクタを差し込む

電源(POWER)をOFFにした、ファミリーコンピュータのエキスパンドコネクタ(機能拡張コネクタ)にキーボードのコネクタを接続します。

②V3カセットを挿入する

V3カセットを、ファミリーコンピュータのカセットコネクタに差し込みます。

③電源をONにする

ファミリーコンピュータの電源(POWER)を入れます。

④BASIC準備よし！

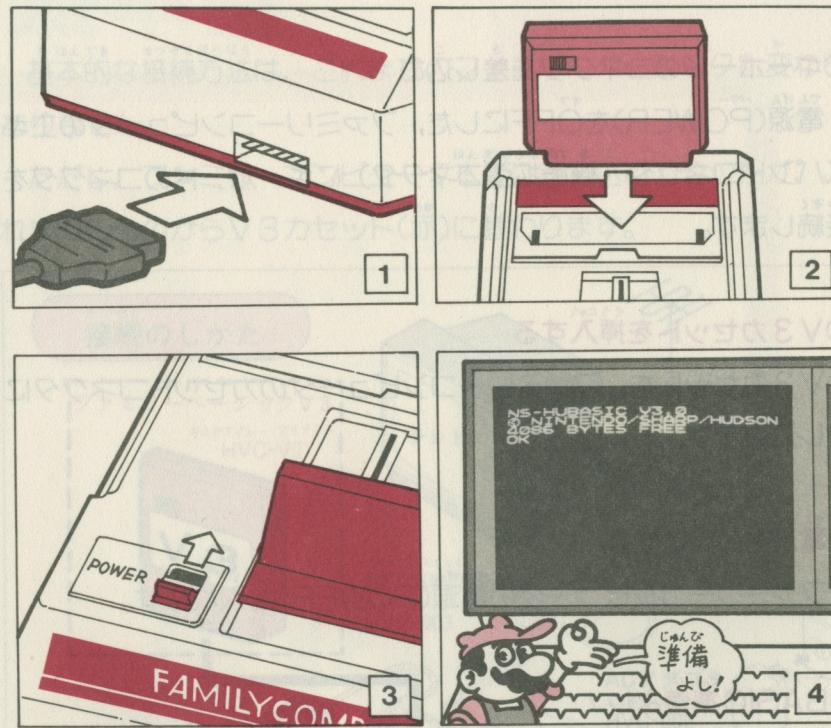
BASICが呼び出され、ファミリーコンピュータの準備が整うと、OKと画面に表示されます。

このとき、OKと表示されずに、

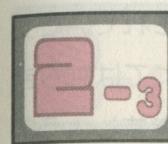
- 1.) 「キーボード ヲ セツゾク シテクダサイ」
とメッセージが出たら、キーボードのコネクタがはずれていないかどうか、もう1度よく確かめてください。また、
- 2.) 「パックアップ スイッチ ヲ OFF ニ シテクダサイ」
とメッセージが出たら、V3カセットの左上部分にあるパックアッ

スイッチが、**ON**になつてないかどうかを確かめてください。

以上の手順を、図にしておきますので参考してください。



このようなスタート方法を、**COLD START**と言います。ふつうは、この**COLD START**で操作を開始します。ただし、この方法はすべて新しい状態ですから、ファミリーコンピュータは何もデータを持っていません。この**COLD START**に対して、**HOT START**という方法があります。この方法は、メモリ/バッファアップのときに使用しますので、詳しくは「2-4. メモリ/バッファアップについて」(P.19)をお読みください。



内蔵プログラムの つか かた 使い方

V3カセットには、4つのプログラムと1つの背景画(BG面)が内蔵されています。これらは、画面にOKと表示されているならば、いつでもプログラムを呼び出すことができます。

プログラムを呼び出す方法は、ファンクションキーを押す方法とコマンドを入力する方法の2つあります。

Ⓐ ファンクションキーで呼び出す方法

4つのプログラムが、ファンクションキー[F1]～[F4]にそれぞれ対応しています。呼び出したいプログラムのキーを押せば、すぐに始めることができます。

Ⓑ コマンドを入力して呼び出す方法

コマンドを入力する場合は、GAME命令を使います。

GAME P,Sw

Pはゲーム番号で、0～3のいずれかを選択します。またSwは背景画をセットから転送するかどうかの指定で、転送する場合は1、しないなら0または省略します。たとえば、GAME0を選択する場合は、

GAME 0 RETURN

とキーを押せばよいのです。



①内蔵プログラムを呼び出すことによって、前に記憶されていたプログラムは自動的に消えてしまいますので、注意しなくてはなりません。残したいときは、カセットテープに保存しておきましょう。

②背景画だけを転送することもできます。その場合は、

GAME [] 1 RETURN

とキーを押します。

③4つの内蔵されているプログラムと1つの背景画は、どんな操作をしても消えることはありません。

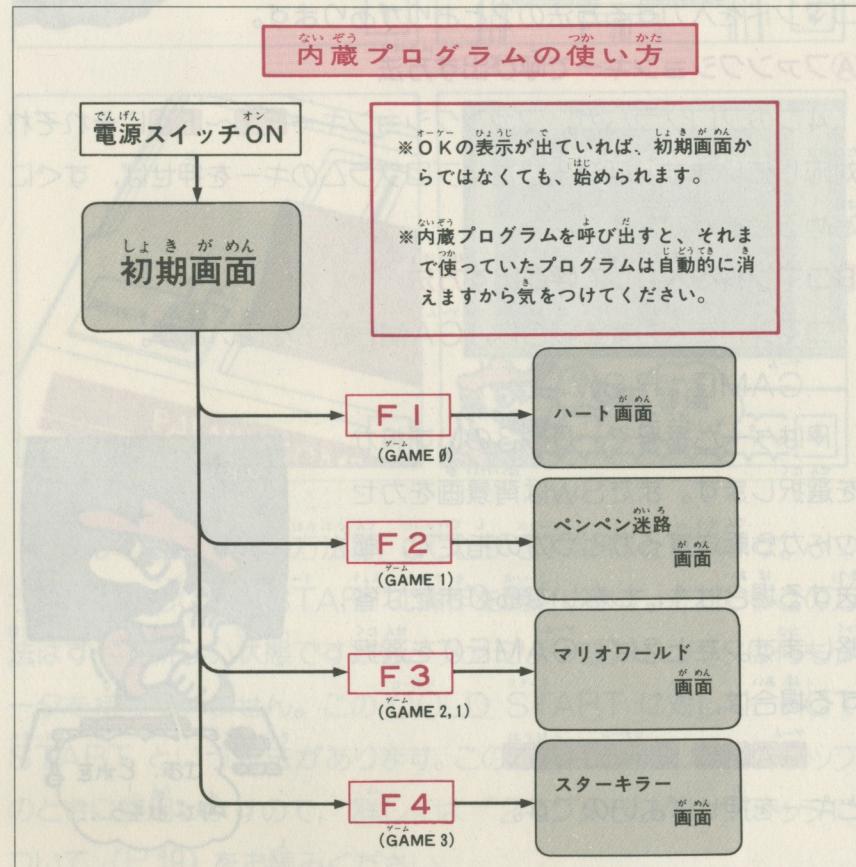


図6

2-4

メモリバックアップについて

BASICのプログラムやBGグラフィックの画面は、単三乾電池を利用して、一時的にV3カセット内に記憶(メモリバックアップ)しておくことができます。プログラムを作っている途中で、電源を切ってひと休みする場合などに利用してください。

A. プログラムとBGグラフィックの両方をメモリバックアップする場合

この場合には、最初にBGGET命令でBGグラフィックを保存する準備をし、その後でBACKUP命令を使い、画面の指示に従ってバックアップスイッチをONにします。BGGET命令の後でBACKUP命令という順番を、まちがえないように気をつけてください。



①BGGET RETURN BACKUP RETURNとキーを押します。

②「バックアップ スイッチ ヲ ON ニ シテクダサイ」とメッセージが表示されます。

③V3カセット左上のバックアップスイッチをON(右側)にします。

④「デンゲン ヲ OFF ニ シテクダサイ」

とメッセージが表示されます。

⑤本体の電源をOFFにし、V3カセットを抜いて終わりです。

なお、BGGET命令を入力すると、“OM ERROR”と出ることがあります。これは、「プログラムと BGグラフィックの両方をバックアップするにはメモリが足りない」という意味です。この場合は、どちらか一方をカセットテープに保存するなどして、残りの片方だけをバックアップしてください。

B. プログラムだけを

メモリバックアップする場合

基本的には両方バックアップする場合と同じですが、BGグラフィックのバックアップ準備が不要となり、手順は次のようにになります。

①[BACKUP] [RETURN]とキーを押します。

②以下の手順は、Aとまったく同じです。

C. BGグラフィックだけを

メモリバックアップする場合



これも、基本的には両方バックアップする場合と同じです。しかし、この場合は、プログラムが不要ということなので、両方バックアップする場合の手順①を次のように変更し、バックアップする前にプログラムを消してしまいます。

この場合、1度消したプログラムは元に戻せませんので、残しておきたいプログラムはカセットテープに保存しておくようにするとよいでしょう。

① NEW RETURN BGGET RETURN BACKUP RETURN

とキーを押します。

②以下の手順は、Aとまったく同じです。

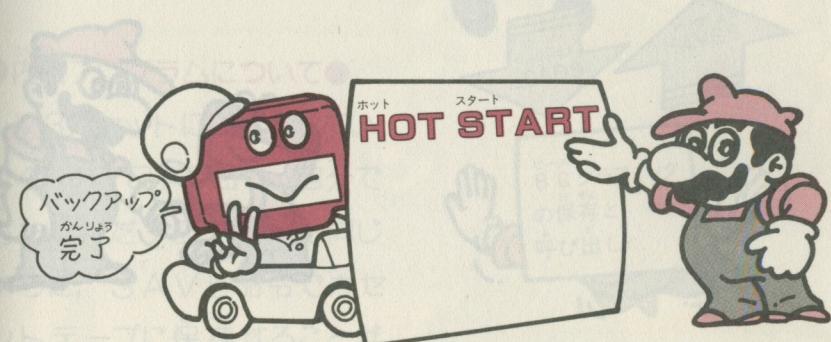
D. メモリバックアップしたものを使える状態に戻す手順

①V3カセットを本体に差し込み、本体の電源スイッチをONにします。

②「バックアップ スイッチ ヲ OFF ニ シテクダサイ」とメッセージが表示されます。

③V3カセット左上のバックアップスイッチをOFF(左側)にします。

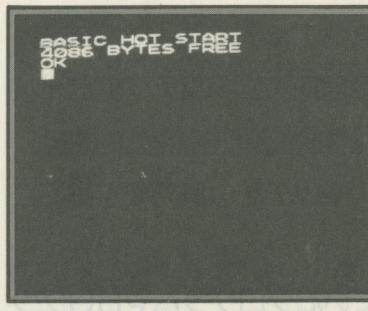
④BASICの状態になり、記憶されていたプログラムやBGグラフィックが使えるようになります。



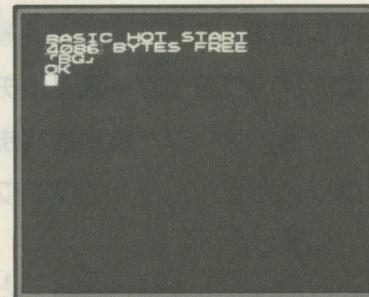
なあ、バックアップスイッチをONにした状態で、本体の電源スイッチをONにしてスタートすることをHOT STARTといい、画面にも「HOT START」と表示されます。これは、すでにプログラムがバックアップされている状態です。

また、BGグラフィックがある場合は、「HOT START BG」と表示されます。そして、自動的にユーザーRAMからVRAMへ、BG面1のデータが転送され、BGPURT命令の実行と同じ状態になります。

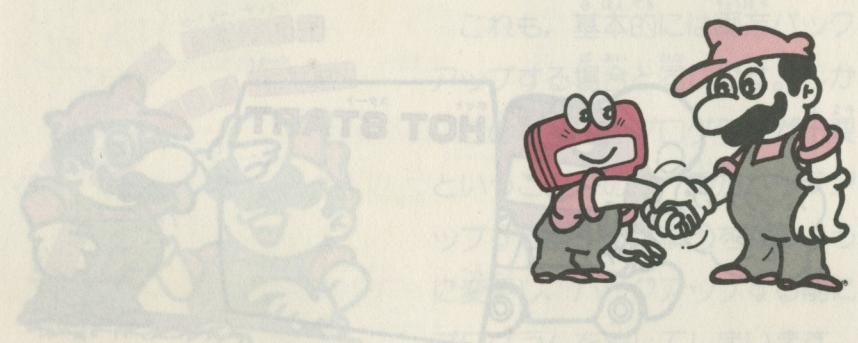
このとき、もう1度BGPURT命令を実行しても、データはすでに転送済みのため、NBエラーが発生します。



HOT START



BG グラフィックがある場合



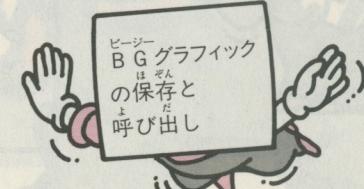
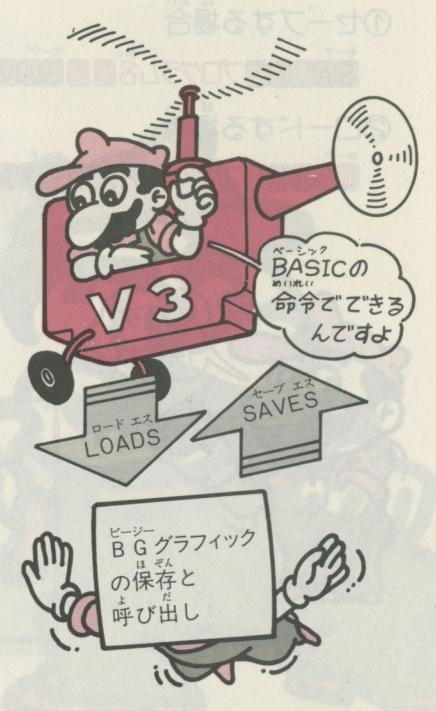
2-5 カセットテープへの保存と呼び出し

ファミリーベーシックV3では、これまでのSAVE/LOAD(プログラムの保存/呼び出し)命令の他に、SAVES/LOADS(BGグラフィックの保存/呼び出し)という2つの命令が追加されました。この命令によって、これまでBGグラフィック作成機能を使わないとできなかつた画面の保存が、BASICの命令でできるようになりました。それぞれの命令の使い方は、プログラムのSAVE/LOADと同じです。(ファミリーベーシック取扱説明書の「プログラムのセーブ(保存), LOAD(呼び出し)」P.45~48)

また、これまでのファミリーベーシックで作つて、カセットテープに保存してあつたBGグラフィックも、この命令を使つて読み込むことができます。

●内蔵プログラムについて●

V3カセットに内蔵されている4つのプログラムも、自分で打ち込んだプログラムと同じように、SAVE命令でカセットテープに保存することができます



できます。もちろん、いろいろな改造をした後で、SAVEすることもできます。

また、**SAVES/LOADS**で画面の保存／呼び出しもできますから、『マリオワールド』の画面をいくつも作っておいて、使い分けるというような応用もできるわけです。

● プログラムとBGグラフィックと一緒にSAVE/LOADする方法

プログラムとBGグラフィックを組み合わせて、1つのゲームになっているものなどは、SAVE/LOADするときも、一緒になつていたほうが便利です。次のようにすれば、プログラムとBGグラフィックを、同時にSAVE/LOADすることができます。

①セーブする場合

セーブ [セーブ] プログラム名 [セーブ] **SAVES** [セーブ] BGのファイル名 [リターン]

②ロードする場合

ロード [ロード] プログラム名 [ロード] **LOADS** [ロード] BGのファイル名 [リターン]



※V3カセットでの、スタートから終了までの全体の流れ、操作手順はP.121~125で図解しています。

ファミリーベーシック

3. FAMILY BASIC V3のゲームプログラム



3-1

プログラムの かいぞうほうほう 改造方法

ファミリーベーシックV3には、ワンタッチで呼び出せる4つのプログラムが内蔵されています。呼び出したプログラムは、そのままでも遊べますし、BASICで作られていますから、自分で改造することができます。

改造で必要となるリストの変更方法を説明しておきましょう。

■必ず、次の操作をした後に、改造を始めてください。

1. [STOP] キーを押してプログラムを止めます。

2. [SHIFT] + [CLR HOME] キーを押すと、リストが消え、カーソルがホーム位置に戻ります。

3. [CTR] + [□] キーを押してアニメキャラクタを消し、文字の色を白色に戻します。

4. リスト命令で、必要なプログラム行を呼び出します。

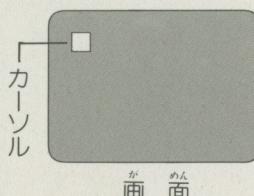
① 1文字変更したいとき

$100 \times=2$ を $100 \times=3$ のように、1文字だけ変更したいときは、

$100 \times=2$

↑
カーソル

カーソルを「2」の上に動かして、③キーを押してから [RETURN] キーを押してください。



画面

② 間に何文字かを入れたいとき

$100 A=SCR$(X, Y)$

を

$100 A=SCR$(\underline{X+1}, Y)$

のように変更したいときは、

$100 A=SCR$(X, Y)$

↑
カーソル

カーソルを「」の上に動かし、INS INS +

① [RETURN] キーを押します。INS キー

を押すと、押した回数だけ間があきます。

③ 何文字かを詰めたいとき

100 PRINT SC+ST

1回押す



100 PRINT S+ST

押した回数だけ、カーソル
が消していきます。



100 PRINT SC+ST

を

100 PRINT S+ST

のように変更したいときは、

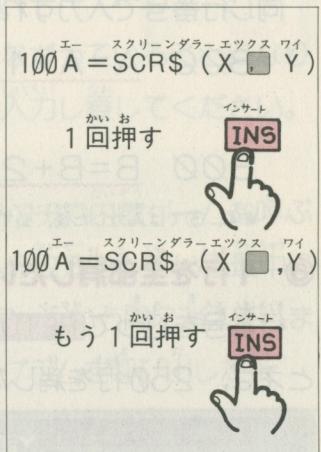
100 PRINT SC+ST

↑
カーソル

カーソルを「+」の上に動かして [DEL] キーを押してください。その後 [RETURN] キーを押します。押した回数分だけ、カーソルが左側の文字を消していきます。

④ 新しい行を追加したいとき

画面に「OK」が表示されて、その下にカーソルが出ているときに、追加したい行だけを行番号を付けて入力してください。入力を終えてからリストを出せば、追加した行も含んで行番号順に出てきます。



⑤ 1行を全部変更したいとき

同じ行番号で入力すれば、あとから入力したものに変わります。

300 A=A+10 : GOTO 500 ←前の行を消さなくても

300 B=B+20 : GOSUB 650 ←同じ行番号で入力すれば、これに変わります。
※④, ⑤では入力した後、必ず RETURN キーを押してください。

⑥ 1行を全部消したいとき

行番号を打って RETURN キーを押せば、その行が消えます。たとえば、250行を消したいならば、

250 RETURN

とキーを押してください。

⑦ 行と行がつながってはいけません

プログラムの中で、ある行と次の行がつながつてしまうと、エラーになつたり、正しく動かないことがあります。

ある行が画面のいちばん右端で終わっているときは、リストを出すと次の行との間が1行あきます。たとえば、

300 IF X<8 AND Y<8 GOTO 1000

1行あく

310 X=X+AX : Y=Y+AY

のように、300行が画面の右端で終わりなら、リストを出したとき、300行の下に1行あいて310行が出てくるのが正しいのです。これが、



300 IF X<8 AND Y<8 GOTO 1000

310 X=X+AX : Y=Y+AY

と出てきたときは、300行と310行は、つながつてことになります。このようなときは、もう一度正しく入力し直してください。

⑧ 内蔵プログラムを改造する前に

F1 F2 F3 F4 の4つのキーは、ワンタッチで内蔵ゲームを呼ぶことができる便利なキーですが、プログラムの改造をしている最中にこのキーを押すと、改造したプログラムがアツという間に消えてしまふことがあります。これを防止するには改造したいプログラムを呼び出した後

キー	KEY 1, " "	リターン
キー	KEY 2, " "	リターン
キー	KEY 3, " "	リターン
キー	KEY 4, " "	リターン
KEYLIST		

KEY 1, " "	SPRITE	写真 I
KEY 2, " "	LOAD CM	
KEY 3, " "	LIST CM	
KEY 4, " "	RUN CM	
KEYLIST		

とキーを押してください。次に、

キー	KEY F7	リスト	リターン
(または KEYLIST RETURN)			

E5	SPRITE	E7	KEYLIST
E6	LOAD	E8	RUN

とキーを押して、写真 I のように、F1 ~ F4 の右側に何も出ないことを確かめてください。こうしておけば、まちがつて F1 ~ F4 のキーを押してしまっても、改造中のプログラムが消えてしまうことはありません。

F1 ~ F4 のキーを元に戻すには、ファミリーコンピュータ本体のリセットキーを押します。ただしリセットキーを押すと本体のプログラムはそのまま残りますが、BGグラフィックは消えてしまいます。BGグラフィックを残したいときは、BGグラフィックをテープにセーブしてから、リセットキーを押してください。

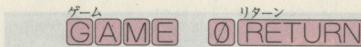
3-2

ないぞう 内蔵プログラムで あそ 遊ぼう

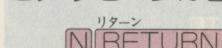
1. ハート

A. あそび方

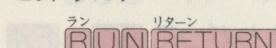
プログラムを呼び出すときは、**F1**キーを押すか、または、



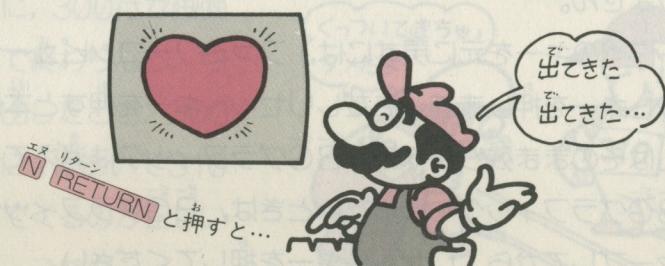
とキーを押してください。プログラムが始まるとき、音楽が流れ、
PLAY ? Y/N=> (YはYes, NはNoの意味です)
とメッセージが出てきます。カーソル



とキーを押してください。こうするとハートがでてきます。遊びたくなつたら **STOP** キーを押してプログラムを止めてから、**F8** キーを押すか。



とキーを押せば、メッセージが再び出でてきます。

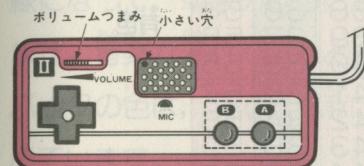


二二四

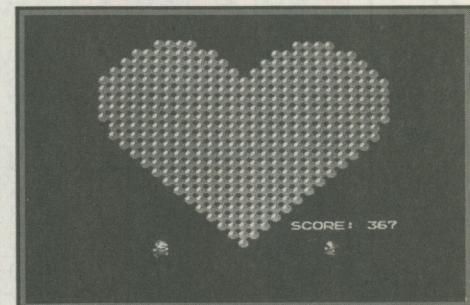
リターン
RETURN

キーを押すと画面は暗くなり、ハートを描く準備ができました。

そこで、IIコントローラのVOLUME(ボリューム)のつまみをいちばん右まで動かしてから、マイクの左上の小さな穴に向かって息を吹きかけるか、声を出してください(図7)。



す 7



しゃしん 写真

息を吹きかけると、ハートのマークが少しづつできてきます。次にマリオとレディーが画面の下に出てきて、真ん中で出合うと音楽が流れます(写真2)。終わるとまた最初のメッセージ画面に戻りますので、くり返して遊べます。終わりたいときは、**STOP** キーを押してください。

B. 改造の方法

① ハートの色を変えてみましょう

えが
いろ
ぎょう
き
ハートを描いているときの色は、80行で決めていきます。

80 スプライト オン シージー セット アンドヘクサー
SPRITE ON · CGSET Ø. Ø · U = 8H15

「&H15」は16進数で、10進数の数字にすると「21」のことです。この数字をほかの数字に変更すると、ハートの色が変わります。

※文中に出てくる「&」は、「&」と同じ文字です。画面に出てくる
る文字と多少異なりますので、ご了承ください。

何色になるのかは、図8の色コード表でています。たとえば「U=39」にすると黄色のハートになります。

52色の色コード										
	16進	10進	16進	10進	16進	10進	16進	10進	16進	10進
青系	00	0	10	16	20	32	30	48	灰色～白色	
	01	1	11	17	21	33	31	49		
	02	2	12	18	22	34	32	50		
	03	3	13	19	23	35	33	51		
	04	4	14	20	24	36	34	52		
赤系	05	5	15	21	25	37	35	53		
	06	6	16	22	26	38	36	54		
	07	7	17	23	27	39	37	55		
	08	8	18	24	28	40	38	56		
緑系	09	9	19	25	29	41	39	57		
	0A	10	1A	26	2A	42	3A	58		
	0B	11	1B	27	2B	43	3B	59		
	0C	12	1C	28	2C	44	3C	60		
	0D	13	1D	29	2D	45				
	0E	14	1E	30	2E	46				
	0F	15	1F	31	2F	47				

音楽が流れているときのハート色は、430行の、
 $430 \text{ W}=\&H34 : \text{FOR } N=\text{0TO1} : \dots \text{以下略}$
 で決まります。「&H34」は、10進数の数字では「52」のことです。
 色は淡いピンクです。この数字も色コード表を見ながら、変えてみま
 しょう。



※文中に出てくる「&」は、「&」と同じ文字です。画面に出てく
 る文字と多少異なりますので、ご了承ください。

② バックの色を変えてみましょう

バックの色は黒ですが、この色は90行のPALET B 命令で決めています。

90 PALET B Ø, 15, 48, 48, U

この数字も色コード表を見ながら
 変えてみましょう。たとえば
 「10」にすると、バックの色が
 緑になって、草原のようになります。
 音楽が流れているときの
 バックの色は、480行で赤に決
 めています。

480 LOCATE1Ø, 1Ø: PRINT "PERFEC
 T" : FOR N=ØTO1ØØ: PALET B Ø, 6, &H
 3Ø,以下略

この「6」も、ほかの数字に変えてみましょう。



バックの色を変えるときは、ハートの色と同
 じにならないように注意してください。ハート
 が見えなくなってしまいます。また、バックの
 色を白にすると、STOPキーを押して中断した
 ときに、文字が見えなくなってしまいます。こ
 のようなときは、

コントロール
 CTR + D

コントロール
 (CTR)キーを押しながら□キーを押します)
 とキーを押せば、文字が見えるようになります。

※文中に出てくる「&」は、「&」と同じ文字です。画面に出てく
 る文字と多少異なりますので、ご了承ください。

③ ハートを描くキャラクタを変えてみましょう

ここでは、ハートを丸いボールで描いていますが、何かほかのキャラクタで描いてみましょう。それには、まずプログラムの3カ所を、次のように変えてください。

```

300 ロケート LOCATE X, Y: PRINT "●" ; : …以下略
300 ロケート LOCATE X, Y: PRINT H$ ; : …以下略
350 X=X+1: IF SCR$(X, Y) = " " AND
    X<=E THEN ロケート LOCATE X, Y: PRINT "
●": GOTO 330
350 X=X+1: IF SCR$(X, Y) = " " AND
    X<=E THEN ロケート LOCATE X, Y: PRINT "
H$": GOTO 330
410 X=-1: FORC=S TOE: IF SCR$(C
    , Y) = "●" AND……以下略
410 X=-1: FORC=S TOE: IF SCR$(C
    , Y) = H$ AND……以下略

```

この3カ所を変えた後、次の1行を新しく追加してRUNしてみましょう。

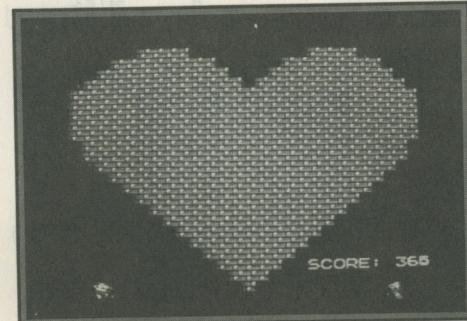


写真3

135 H\$=CHR\$(195)
すると、ハートガレンガで描かれるようになります(写真3)。

新しく追加した135行の
CHR\$()の中の数字を変えると、いろいろなキャラク

タでハートを描くことができます。184~255の間の数字にすれば、BGグラフィックで使うキャラクタになり、33~183にすると文字や記号になります。どのキャラクタになるのかは、本書付録(P.134~135)のキャラクタコード表Bを見てください。なお、キャラクタの種類によつては、色が変わらないことがあります。

④ タイマーを付けてみましょう

ハートのプログラムは、PERFECTの文字が表れるまで実行されますが、タイマーを作つて制限時間を設けましょう。

タイマーを付けるためには、いくつかの行を変えたり、追加しなければなりません。

次のように行を変えてください)

```

230 IF (PEEK (&H4016) AND &H4)
    =Q THEN 230
230 IF (PEEK (&H4016) AND &H4)
    =Q THEN 225
330 IF (PEEK (&H4016) AND &H4)
    =Q THEN 330
330 IF (PEEK (&H4016) AND &H4)
    =Q THEN 325

```



※文中に出てくる「&」は、「&」と同じ文字です。画面に出てくる文字と多少異なりますので、ご了承ください。

440 IF (PEEK (&H4016) AND &H4)
 =Q THEN440
 440=IF (PEEK (&H4016) AND &H4)
 =Q THEN435

（次に新しい行を追加してください）

- 145 DEF MOVE (7) = SPRITE (11, 3, 25, 255, 0, 0) : POSITION 7, 16, 20 : MOVE 7
- 225 IF XPOS (7) > 216 AND Q <> 99 9 THEN P=L:NEXT:RETURN 600
- 325 IF XPOS (7) > 216 AND Q <> 99 9 THEN RETURN 600
- 435 IF XPOS (7) > 216 AND Q <> 99 9 THEN RETURN 600
- 485 CUT 7
- 600 , ジカンギレ
- 610 CGSET1, 1:CUT 7:PLAY "02T1 B3AGFEDC"
- 620 LOCATE 10, 7:PRINT " ジカ ンギレ "
- 630 PAUSE 300:ERA 7
- 640 GOTO 190

※文中に出てくる「&」は、「&」と同じ文字です。画面に出てくる文字と多少異なりますので、ご了承ください。



このように改造すると、画面の上方に二タニタのタイマーが出てきます。はじめ二タニタは左の方にいて、少しずつ右に動いて行き、いちばん右まで行くと、時間ぎれになつて終わります。

二タニタの動く速さを変えると、制限時間を変えることができます。
新しく追加した行のDEF MOVE 命令の中に

145 DEF MOVE (7) = SPRITE (11, 3,

25, 255, 0, 0) :以下略
25という数字があります。この数字を大きくすると制限時間が長くなり、小さくすると短くなります。でもあまり大きくしたり小さくしたりすると、タイマーの意味がなくなつてしましますから注意してください。

C. マイクを使ってみましょう

① マイクの使い方



※文中に出てくる「&」は、「&」と同じ文字です。画面に出てくる文字と多少異なりますので、ご了承ください。

IIコントローラにはマイクが付いています。このマイクに音や声が入ったかどうかは、BASICで知ることができます。これには、PEEK()命令とメモリ&H4016番地を使用します。では、次のような1行のプログラムを入力して、RUNしてください。

1 PRINT PEEK(&H4016) AND&H4: G
OTO 1

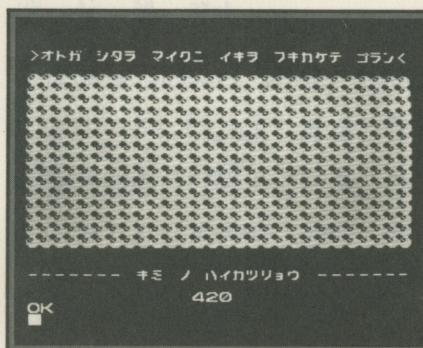
RUNすると、画面に数字が次々と出てきます。そこで、VOLUMEをいっぱいにしてからマイクに向かって息を吹いたり声を出すと、その数字が変わります。数字が変わったときが、マイクに音が入ったときなのです。



これをプログラムで使うには、はじめに PEEK(&H4016) AND&H4 を変数に代入しておき、マイクからの音入力で PEEK(&H4016) AND&H4 の値と同じかどうか、判定します。このテクニックを使ったプログラムの例を次にあげておきましょう。

② 肺活量遊び

このプログラムは、一息がどのくらいの長さになるかを数字で表しています。RUNするとメッセージが出て、スタートの音がします(写真4)。



すぐにマイクに向かって息を吹いてください。息がとぎれると、そこで数字が出てきます。大きな数字にしたいのなら、そうつと長く吹きかけるのがコツです。もう一度始めたいときは、RUNしてく

ださい。ここに出てくる数字は学校などで測る肺活量と同じではありません。また、画面に出てくるBGキャラクタの種類と色は、RUNのたびに変わります。

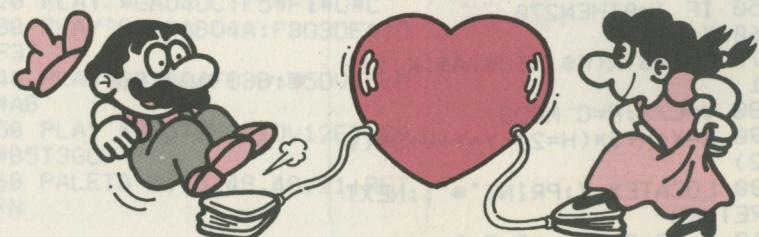
```

100 'ハイカツリョウ ケーム
110 CLEAR:CLS:C=0:F=0:CP=184
+RND(72)
120 PV=PEEK(&H4016)AND&H4
130 PALETB 0,14,48,48,17+RND
(12)
140 PRINT ">オトカ" シラマイクニ イキ
ヲフキカケテ コ"ラン"
150 PAUSE 200 : PLAY "V1503T
1C1DEFGAB04C"
160 IF (PEEK(&H4016)AND&H4)= ← 息が吹きかけられるま
PV GOTO 160
170 FOR SS=0 TO 19
180 S=PEEK(&H4016)AND&H4
190 IF S<>PV THEN SS=19:F=1
200 NEXT
210 IF F<>1 GOTO 240
220 C=C+1:PRINT CHR$(CP);
230 IF S<>PV THEN F=0:GOTO 1
70
240 PRINT:PRINT
250 PRINT "----- キミ ノ ハイカツ
リョウ -----"
260 PRINT "----- ■■■;C;"
270 END

```

※ カナモードで、[GRPH]+[ヘル]を押すと
■になります。

グラフィック + ヘリ



ハート プログラムリスト

```

10 PLAY "M1V9Y2T3:M1V7Y1T3:M1
T3"
20 PLAY "03E5G:03C5C:02G5G"
30 PLAY "04C03B4A1:CC:#F#F"
40 PLAY "G4A1:F4F1:03D402B1"
50 PLAY "B404C7:F4E7:G4C7"
60 CLS:LOCATE 8,10:Q=PEEK(&H
4016) AND &H4:INPUT "PLAY ? Y
/N>",Y$:IF Y$="N" THEN Q=99
9
70 CLS:ERA0,1,2,3,4,5,6,7
80 SPRITE ON:CGSET0,0:U=&H15

90 PALETB 0,15,48,48,U
100 A$="AIIIIIIIIIIIGIGGGGH
"

110 B$="HHBHBEEBEEE"
120 K$="ABCDEFGH"
130 I$="ABCGHIDEF"
140 CLS:LOCATE9,10:PRINT"STA
RT !":PAUSE80:CLS
150 X=13:Y=22:I=0
160 GOSUB220:SWAP A$,B$:GOSUB
220
170 IF I=0THEN SWAP K$,I$:=I:=
GOTO160
180 GOSUB310:GOSUB430
190 IF Q<>999THEN60
200 GOT070
210 END
220 L=LEN(A$):FOR P=1 TO L
230 IF (PEEK(&H4016) AND &H4)
=Q THEN230
240 K=P
250 IF I=0THEN270
260 K=L+1-P
270 C=INSTR(K$,MID$(A$,K,1))
-1
280 V=C/3:H=C MOD3
290 X=X+H+3*(H=2):Y=Y+V+3*(V
=2)
300 LOCATE X,Y:PRINT "*";:NEXT
:RETURN
310 Y=22:S=0:E=13:T=0

```

●音楽を流して遊ぶかどうかを選ぶ

●画面の色を決めるなどの準備

●メインルーチン
(サブルーチンを呼ぶ)

●ハートのふちを描く

※本書リストに出てくる「*」は、「*」と同じ文字です。画面に出てくる文字と、多少異なりますので、ご了承ください。

```

320 LOCATE18,20:PRINT"SCORE:
";
330 IF(PEEK(&H4016) AND &H4)
=Q THEN330
340 T=T+1:LOCATE24,20:PRINTT
350 X=X+1:IF SCR$(X,Y)=" " AND
X<=E THEN LOCATE X,Y:PRINT "
":
360 Y=Y-1:IF Y<1 THEN390
370 GOSUB410 :IF X=-1 THEN360
380 GOTO350
390 IFS=0THENS=13:E=26:Y=20:
GOSUB410:GOTO330
400 RETURN
410 X=-1:FOR C=S TO E:IF SCR$(C
,Y)=" " AND SCR$(C+1,Y)=" " TH
EN X=C:C=E
420 NEXT:RETURN
430 W=&H34:FOR N=0 TO 1:DEFMOVE
(N)=SPRITE(N,3+4*N,1,1,0,0):
POSITION N,236*N,200:NEXT
440 IF(PEEK(&H4016) AND &H4)
=Q THEN440
450 T=T+1:LOCATE24,20:PRINTT
460 MOVE 0,1:IF CRASH(0)=1 THEN
480
470 GOTO440
480 LOCATE10,10:PRINT"PERFEC
T":FOR N=0 TO 100:PALETB 0,6,&H
30,&H30,U:SWAP U,W:NEXT
490 PLAY "M1V2Y1T2:M1V10Y1T2:
T2"
500 PLAY "03E3G#DGDG#DG:01#B5
B#AB:R"
510 PLAY "EG04C03E:Y202#B6E3:
01C5C"
520 PLAY "#GA04DC:F5#F:#C#C"
530 PLAY "03B#AB04A:F303DFA:D
6F3"
540 PLAY "G03A04F03B:E5DV15:G
A#AB"
550 PLAY "#B3GT3FGE1DV12E7:C9
:#B5T3GC7"
560 PALETB 0,15,48,48,21:RET
URN

```

●ハートを塗りつぶす

●マリオとレディを動かす

●マリオとレディが出会ったとき
音楽を流す

ハート変数リスト

Q	ゲームをするかしないか
X, Y	BGキャラクタを描くBG座標
U	ハートの色
W	ハートが完成し音楽が鳴っているときのハートの色
I, L, P, K, C, V, H	ハートを描く作業用
A\$, B\$, K\$, I\$	ハートの形をあらわす文字データ(注)参考
N	左右から出てくるアニメキャラクタのMOVE番号
T	進度カウンター
(注) 文字データは、次のようにX, Yの変化分を表す。	
Yの変化分	(0)
Xの変化分	(+1)
文字データ	(-1)
	0, +1, -1
	A, B, C
	0, +1, -1
	D, E, F
	0, +1, -1
	G, H, I

2. ペンペン迷路

A. 遊び方

プログラムを呼び出すときは、F2キーを押すか、または、
GAME リターン
 とキーを押してください。ゲームが始まると、写真5のような画面になります。この画面に見えている縦・横の線が迷路です。ペンペ

ンはこの上を動きまわることができます。

迷路の改造方法は、あとで説明します。

ゲームを始めるには、IコントローラのAボタンを押します。

ゲームが始まるとペンペンが出てきますから、IコントローラのAボタンを押して迷路に沿って動かしてください。迷路には1~9の数字があります(写真6)。それを小さい順に1個通過するごとに、10点ずつの得点になります。全部取れると

フェクト!!

ところが、迷路にはカニさんがないて、ペンペンを追いかけ来ます。

カニさんにつかまるとゲームは終わりです。つかまらないようにうまく逃げてください。ペンペンはワナを3つ持っていて、Aボタン

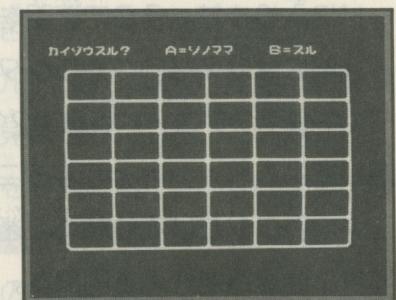


写真5

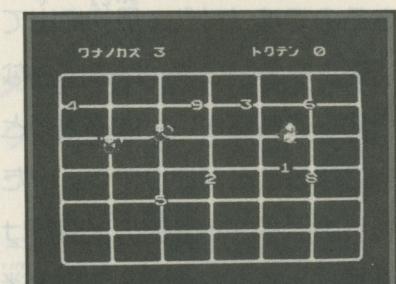


写真6

を押して迷路におくことができます。ワナにかかったカニさんは、しばらくの間動きなくなります。ゲームが終わると、「オシマイ」のメッセージが出ます。再びゲームを始めるには、①コントローラのSTARTボタンを押してください。それ以外のボタンを押すと「OK」が出て、プログラムは終了します。

B. 改造の方法

ペンペン迷路ゲームは、BGグラフィックをいろいろと変えて遊べます。ゲームの始まりでBボタンを押すと、BGグラフィックモードに入れます。そこで迷路を変えてみましょう。

迷路には使えないキャラクタが2種類あります。ワナのキャラクタ(キャラクタコード表Bの219)と、BGグラフィックモードで点滅している「へ」(キャラクタコード表Bの94)です。

キャラクタだけではなく、迷路の形も変えることができます。

写真7は迷路を変えた例です。このように自由にデザインすることができます。ただし、最初に出てきた迷路より外側には、はみ出さないようにしてください。はみ出すとエラーになることがあります。また、**写真5**のいちばん外側の迷路だけは、必ず作ってください。外側の迷路がないと、デザインした迷路が消えてしまったり、ペンペンが動けなくなってしまうことがあります。

気に入ったBGグラフィックができたら、BGグラフィックモードで「FILE」を選び、またはBASICのSAVE S命令を使って

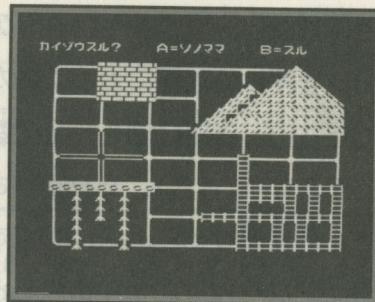


写真7

テープにセーブしておけば、あとでその画面を使うことができます。テープにセーブしたBGグラフィックを使うときは、BGグラフィックモードで「FILE」を選び、BASICのLOADS命令で本体にロードします。「FILE」を選べば、ロードした絵をすぐに見ることができます。

① 色を変えてみましょう

迷路の色やバックの色を変えたいときは、新しく1行追加します。

105 PALETB Ø, 10, 48, 19, 37

この数字でRUNすると、バックの色が緑、迷路の色がピンク、また、「オシマイ」のメッセージの枠が白と紫になります。この色を変えたいときは、図8の色コード表を見ながら、いろいろな数字にしてみましょう。色を変えるとゲームをやめても、その色のままになっています。元に戻したいときは、CTRキーを押しながら□キーを押してください(色表示はテレビの機種によって多少異なる場合があります)。

② ペンペンやカニさんをほかのアニメキャラクタに変えてみましょう

ペンペンを、ほかのアニメキャラクタに変えたいときは、390行と580行を変更してください。

390 DEF MOVE(7)=SPRITE (4, Ø, ...以下略)

580 DEF MOVE(7)=SPRITE (4, S, ...以下略)

4を0~15の数字にすると、ペンペン以外のアニメキャラクタになります。どのアニメキャラクタになるのかは、本書の付録P.128(下段)をご覧ください。

たとえば、BGグラフィックで迷路をハシゴにしたときは、この数字を0にして、マリオを出すと自然です。

カニさんを変えたいときは、次の3つの行の14を同じように変更してください。



```
430 DEF MOVE (F)=SPRITE (14, 1, ...
820 DEF MOVE (F)=SPRITE (14, 1...
1070 DEF MOVE (F)=SPRITE (14, ...
```

③ 得点を変えてみましょう

このゲームでは、数字を1つ拾うと10点ですが、この得点を変えてみましょう。

```
630 T=T+10: LOCATE.....以下略
```

得点は上のように、630行の10で決めています。この10を20や100にすると、数字を拾ったときの得点が変わります。あまり大きな数字にすると、エラーになることがあります。

④ ワナの数を変えてみましょう

ワナの数は360行で決めています。ワナの数を変えたいときは、360行の3をほかの数字にします。

```
360 PX=13*8+12: PY=2*8+16: C=3
:T=0: FW=0: F=0
```

プレイ中にワナを置くときは、前に置いたワナが消えるまで、次のワナを置くことができません。

⑤ カニさんの数を変えてみましょう

KNがカニさんの数を表しています。370行にKN=2というのがありますから、この数字を変えてみましょう。

370 KN=2: W=219

ただし、アニメキャラクタは、5つ以上が横に並ぶと、5つめは画面から消えてしまいます。カニさんの数は3にしておないと、どれか1つ見えなくなってしまうことがあります。カニさんの数を0



以下にすると、エラーになってしまいます。ですから、カニさんの数は1か2か3にしておいてください。

⑥ ワナをうまくおくことができないときは……

ワナをうまくおくことができなかつたり、ワナを使った後に迷路があかしくなったときは、555行をそのまま490行にしてください。この方法は、まず、

リスト 555 RETURN

で555行を画面に出します。

そしてカーソルを左端の5の上に動かして、行番号だけを490と打ち直し、RETURNキーを押します。

これで555行が490行にコピーされました。あとは、

リターン
555 RETURN

とキーを押して、555行を消してしまいます。

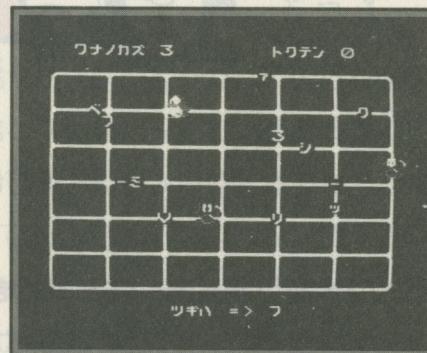
ある行をほかの行に移したいときは、このようにします。自分でプログラムを作るときなどに便利ですから、この方法を覚えておくとよいでしょう。

⑦ ペンペンが拾えるものを数字以外のキャラクタにしてみましょう

ペンペンは数字の1～9を、順番に拾えるようになっています。
この順番や個数を変えたり、数字以外のアルファベットやカタカナ
にもできます。これらは290の行で変更できます。

290 PS\$="123456789":PL=.....以下略

290行を見ると、PSS\$とい
う文字変数に "123456
789" を代入しています。
この " " ではさまれた中身
が拾えるキャラクタです。こ
の中に書いた順番に、書いた
個数だけを拾えるようにでき
ます。たとえば:



しゃしん
写真 8

"98765"

"ABCDEFGHIJKLMN"

"MARIO"

“ファミリーベーシックV3” (写真8)

などにしてあけば良いのです。ただし、" " の間に何も書かなか

つたり、書いた文字が31文字を越えると、エラーになりますから注意してください。また、数字はすべて1桁で、“12132”のように、同じ文字が2回以上あってもかまいませんが、“11232”のように、同じ文字を続けて書いてはいけません。

⑧ プレイ中に次に何を拾うかを画面に出してみましょう

新しく2行を追加すると、次に拾うキャラクタを、画面に出してあくことができます。こうするとゲー／＼が進めやすくなります。

335 ロケート LOCATE 10, 22: PRINT "ソギハ =
> " ; LEFT\$ (PSS\$, 1) ;
645 ロケート LOCATE 17, 22: PRINT MID\$ (P
S\$, PS, 1) ; プリント ミッドダラー
ン (写真8)

(写真8)

⑨ ボーナス得点をつけましょう

特別のキャラクタを
拾うと、ボーナス得点
がもらえるようにして
みましょう。

まず、290行の“なか”の中からボーナスキャラクタを1種類決めます。たとえば拾うものが、

"ABXC12X3XZ"

なか き
だとして、その中から「X」を、ボーナスキャラクタにしようと決



めたとします。290行の最後に、次のように付け足します。

```
290 PS$= "ABXC12X3XZ": PL=LEN(PS$)
      : PS=1: B$= "X"
      ↑
      付け足す ボーナスキャラクタをかく
```

次に新しく605行を追加します。

```
605 IF S$=B$ THEN T=T+100
      ポーナス点
```

こうしておくと、ボーナスキャラクタ「X」を拾ったときにだけ、10点のほかに、ボーナス得点100点がもらえるようになります。ボーナス得点も、欲張ってあまり大きくしてはいけません。

得点の合計が、32767を越えるとエラーになります。

⑩ ボーナスキャラクタを拾うとカニさんが逃げるようにしましょう

ボーナスキャラクタを拾うと、カニさんが逃げて行くようにしてみましょう。ボーナスキャラクタの決め方は、ボーナス得点のときと同じで、290行に次のように付け足します。



```
290 PS$= "12B34B43B21": PL=LEN(PS$)
      : PS=1: B$= "B"
      ↑
      付け足す ボーナスキャラクタ「B」
```

次に新しく606行と785行を追加します。

```
606 IF S$=B$ THEN DEF MOVE(6)
      =SPRITE(11, 0, 1, 255, 0, 0): POSITION 6, 116, 10: MOVE 6
      785 IF MOVE(6) <> 0 THEN JX=-J
      X: JY=-JY
```

このように改造すると、ボーナスキャラクタを拾ったときには、画面の上の方に、ニタニタが出てきて笑いだします。そして、今まで追いかけてきていたカニさんが、反対に逃げ始めます。ニタニタが笑っている間は、カニさんが逃げ続けます。ニタニタが笑うのをやめると、カニさんはまた追いかけてきます。ニタニタが笑っている時間は、606行の DEF MOVE で決まります。

```
606 IF S$=B$ THEN DEF MOVE(6)
      =SPRITE(11, 0, 1, 255, 0, 0) ……以下略
```

この 255 を、もっと小さな数字に変えてみましょう。でも、あまり小さな数字にすると、カニさんが逃げるまもなく、ニタニタが笑うのをやめてしまいますから、小さくても 50 くらいにしておいたほうがよいでしょう。

⑪ タイマーを作つてみましょう

これまで、ペンペんに有利な改造が多かったので、今度はタイマーを作つて、ある時間が過ぎるとゲームが終わるようにしましょう。タイマーは、3行追加するだけで作ることができます。



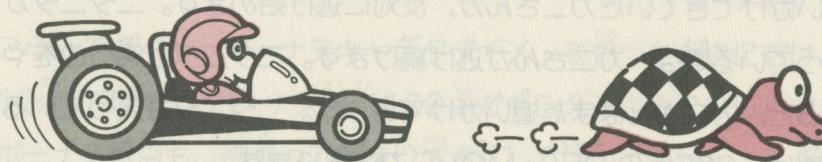
自動車
カメ

```

465 デファン ムーブ   スプライト
      [5, 128, ∅, ∅] : POSITION 5, 116, 21
      ∅ : MOVE 5

466 デファン ムーブ   スプライト
      [ , 128, ∅, ∅] : POSITION 4, 16, 210:
      ムーブ MOVE 4

475 IF クラッシュ GOTO 1120
  
```



このタイマーは、自動車とカメさんの追いかけっこタイマーです。この3行を追加してRUNすると、ゲームが始まつたときに、画面の下の真ん中にカメさんが出てきて、右に向かって歩き始めます。そのカメさんを、左の方から自動車が追いかけてきます。自動車がカメさんに追い付くと、時間切れということになります。ゲームが終わります。

タイマーを作るときには注意が必要です。それは、拾うキャラクタの個数によって、どのくらいの時間で自動車がカメさんに追い付くかを、変えなければならぬからです。たとえば、拾うキャラクタが3つしかないのに、時間がたっぷりあつたのでは、少しもおもしろくありません。拾うキャラクタの個数によって、465行と466行の数字を変えてください。(この数字が速さを決めていきます)

(465 デファン ムーブ スプライト
 [128, ∅, ∅] :以下略

466 デファン ムーブ スプライト

128, ∅, ∅) :以下略

この例では、拾うキャラクタが12個くらいで、ちょうどよくなる速さです。拾うキャラクタがもっと多ければ、両方の数字も少し大きくなり、少なければ小さくなります。速さに応じて、何回か数字を変えながら、ゲームをしてください。数字を変えるときは、カメさんの数字が自動車の数字の、1.8倍くらいになるようにしてください。この例では、

自動車の数字 \times 1.8 = カメさんの数字

25 \times 1.8 = 45

になっています。こうしておけば、画面の右端で、自動車がカメさんに追い付きます。

⑫ カニさんの動きを変えてみましょう。

迷路が、写真9のように切れているところでは、ペンペンは通れないのに、カニさんだけは平気で通つていきます。カニさんも通りないようにするには、次の

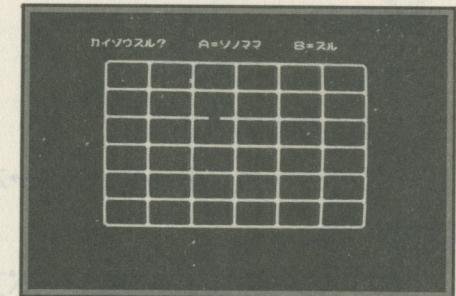


写真9

880 IF (KD=3 AND JX=1) OR (KD=7
AND JX=-1) OR (KD=5 AND JY=1)

OR (KD=1 AND JY=-1) THEN 1060

変更するのは、880行の数字1カ所だけです。「1060」を、「940」にしてください。これで、写真9のように迷路が切れているところは、カニさんも通れなくなります。

⑬ カニさんが追いかけてくる速さを変えてみましょう

カニさんが、ペンペンを追いかけてくる速さは、あまり速くはありませんので、ペンペンは簡単に数字を全部拾うことができます。そこで、プログラムを少し変えると、とても難しいゲームになります。こうするには、1行を変更してから、別に新しい1行を追加します。

<変更する>

```
(700 F=F+1: IF F>KN-1 THEN F=0
700 FOR F=0 TO KN-1
```

<追加する>

1105 NEXT

こうすると動く順番が、

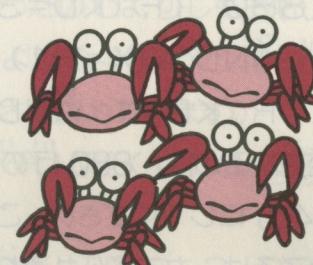
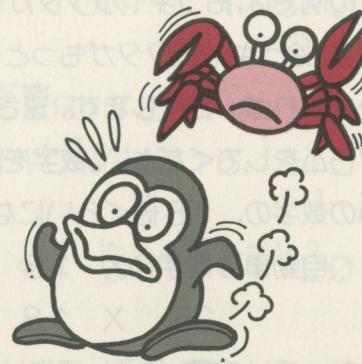
ペンペン→カニ1→ペンペン→カニ2→ペンペン→カニ1……
だったのが、

ペンペン→カニ1→カニ2→ペンペン→カニ1→カニ2

の順番になり、カニさんが速く追いかけてくるようになります。

この改造をすると、ゲームが難しくなりますから、ワナの数を増やしたり、ボーナスキャラクタを、決めておいたほうがよいでしょう。

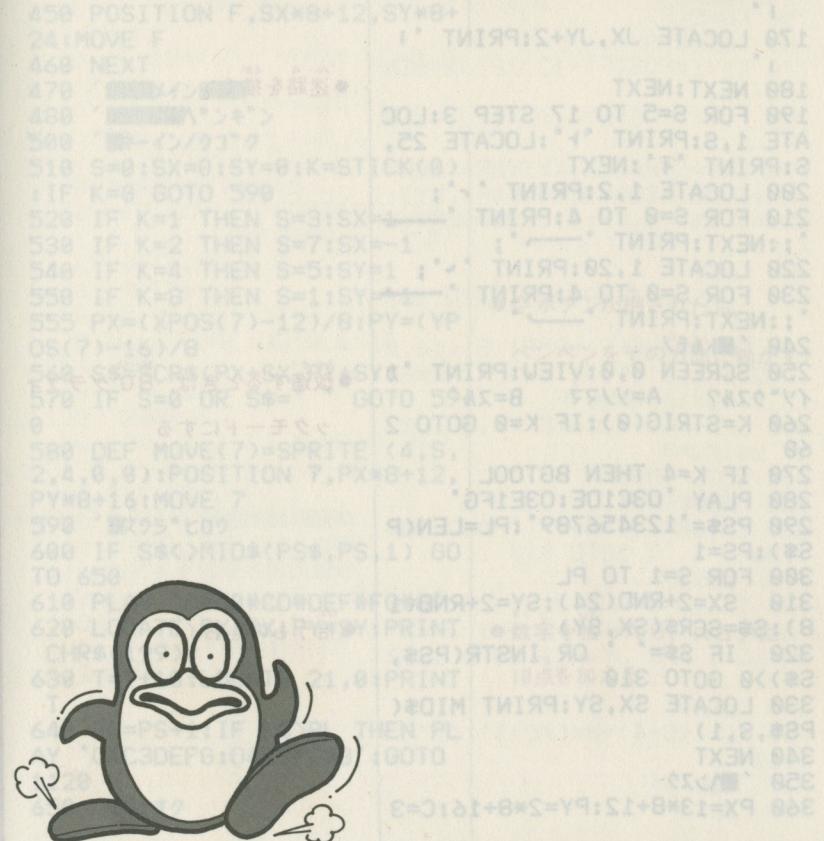
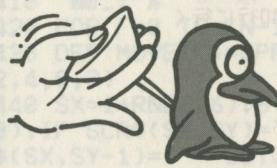
もし、カニさんの数を3以上になると、難しくてゲームになません。



また、ペンペンの動きがぎこちなくなりますので、気になる人は、580行を次のように変えてください。

```
580 DEF MOVE(7) = SPRITE(4, S,
2, 4, 0, 0) : .....以下略
```

「2」を「3」に変えると、少しなめらかになります。



ペンペン迷路プログラムリスト

```

100 CLEAR:CGEN 2:CGSET 1,1:S
CREEN 0,1
110 SPRITE ON:PLAY "V15T1:V1
5T1"
120 IF SCR$(1,2)<>" " GOTO 2
40
130 '■メロ
140 FOR JX=1 TO 21 STEP 4:FO
R JY=2 TO 20 STEP 3
150 LOCATE JX,JY:PRINT "←
+":IF JY>19 GOTO 180
160 LOCATE JX,JY+1:PRINT "↑
↓"
170 LOCATE JX,JY+2:PRINT "↑
↓"
180 NEXT:NEXT
190 FOR S=5 TO 17 STEP 3:LOC
ATE 1,S:PRINT "↑":LOCATE 25,
S:PRINT "↓":NEXT
200 LOCATE 1,2:PRINT "↖";
210 FOR S=0 TO 4:PRINT "→
";:NEXT:PRINT "↖";
220 LOCATE 1,20:PRINT "↖";
230 FOR S=0 TO 4:PRINT "→
";:NEXT:PRINT "↖";
240 '■トルモノ
250 SCREEN 0,0:VIEW:PRINT "カ
イソ"ウスル? A=ソノママ B=スル"
260 K=STRIG(0):IF K=0 GOTO 2
60
270 IF K=4 THEN BGTOOL
280 PLAY "03C1DE:03E1FG"
290 PS$="123456789":PL=LEN(P
S$):PS=1
300 FOR S=1 TO PL
310 SX=2+RND(24):SY=2+RND(1
8):S$=SCR$(SX,SY)
320 IF S$=" " OR INSTR(PS$,
S$)>0 GOTO 310
330 LOCATE SX,SY:PRINT MID$(P
S$,S,1)
340 NEXT
350 '■ハヌク
360 PX=13*8+12:PY=2*8+16:C=3

```

●ゲームの準備

●すでにBGが描いてあれば
240行に行く

●迷路を描く

●改造するときは、BG グラフィ

ックモードにする

●拾うものをおく

56 本書リストに出てくる「*」は、「＊」と同じ文字です。画面に出てくる文字
と、多少異なりますので、ご了承ください。

```

:T=0:FW=0:F=0
370 KN=2:W=219:KX=0:KY=0:KD=
0
380 LOCATE 0,0:PRINT " ワナノ
カス" ;C; " トクテン";T
390 DEF MOVE(7)=SPRITE (4,0,
2,4,0,0):POSITION 7,PX,PY:MO
VE 7
400 DIM E(KN-1)
410 '■カニテル
420 FOR F=0 TO KN-1
430 DEF MOVE(F)=SPRITE(14,1,
2,4,0,3)
440 SX=1+RND(26):SY=10+RND(1
0):IF SCR$(SX,SY)=" " OR SCR
$(SX,SY-1)=" " GOTO 440
450 POSITION F,SX*8+12,SY*8+
24:MOVE F
460 NEXT
470 '■メイン
480 '■パンキン
500 '■イーン/ウコウ
510 S=0:SX=0:SY=0:K=STICK(0)
:IF K=0 GOTO 590
520 IF K=1 THEN S=3:SX=1
530 IF K=2 THEN S=7:SX=-1
540 IF K=4 THEN S=5:SY=1
550 IF K=8 THEN S=1:SY=-1
555 PX=(XPOS(7)-12)/8:PY=(Y
POS(7)-16)/8
560 S$=SCR$(PX+SX, PY+SY)
570 IF S=0 OR S$=" " GOTO 59
0
580 DEF MOVE(7)=SPRITE (4,S,
2,4,0,0):POSITION 7,PX*8+12,
PY*8+16:MOVE 7
590 '■アシ"ヒロウ
600 IF S$<>MID$(PS$,PS,1) GO
TO 650
610 PLAY "04C0#CD#DEF#FG#G"
620 LOCATE PX+SX, PY+SY:PRINT
CHR$(199);
630 T=T+10:LOCATE 21,0:PRINT
T
640 PS=PS+1:IF PS>PL THEN PL
AY "04C3DEFG:04E3FGAB":GOTO
1120
650 '■カナオク

```

●変数を定義してペンペンを出す

●カニさんを出す

●+ボタンが押されたら、
ペンペンをその方向に動かす

●数字を拾ったら、スコアに
10点を加える

```

660 IF STRIG(0)<>8 OR INSTR(
PS$,S$)<>0 OR C<1 OR FW=1 GO
TO 690
670 PLAY "02E2C":W$=S$:LOCAT
E PX+SX,PY+SY:PRINT CHR$(W)
680 FW=1:C=C-1:LOCATE 8,0:PR
INT C
690 ' ■ヒビ
700 F=F+1:IF F>KN-1 THEN F=0 →

710 ' ■シビ"レテル?
720 KD=VCT(F):IF KD>0 GOTO 7
60
730 IF MOVE(F)<>0 THEN PLAY
"00C0EGCEG":GOTO 1080
740 ' ■シビ"レオワリ
750 KD=E(F)
760 ' ■サ"ヒウ
770 KX=(XPOS(F)-12)/8:JX=SGN
(PX-KX)
780 KY=(YPOS(F)-16)/8:JY=SGN
(PY-KY)
790 SK$=SCR$(KX,KY):SR$=SCR$(
KX+1,KY):SL$=SCR$(KX-1,KY):
SD$=SCR$(KX,KY+1):SU$=SCR$(K
X,KY-1)
800 ' ■ワナ?
810 IF SK$<>CHR$(W) GOTO 850

820 DEF MOVE(F)=SPRITE (14,0
,1,200,1,2):POSITION F,KX*8+
12,KY*8+16:MOVE F
830 E(F)=KD:LOCATE KX,KY:PRI
NT W$:FW=0
840 GOTO 1080
850 ' ■オイケル
860 IF(SR$+SL$)<>" " AND SD$+
SU$<>" " GOTO 910
870 ' ■ショクシン
880 IF(KD=3 AND JX=1)OR(KD=7
AND JX=-1)OR(KD=5 AND JY=1)
OR(KD=1 AND JY=-1) GOTO 1060

890 ' ■ハンテン
900 KD=(KD+4)+8*(KD>3):GOTO
1060
910 ' ■フンキ
920 ' ■イケル

```

- Aボタンが押されたら ワナをおく
- 2匹のカニさんのどちらを動かすかを決める
- カニさんがワナでシビレていればそのまま、シビレ終わっているたらワナからぬけ出す
- カニさんのスプライト座標をキャラクタ座標に変えて、ペンペンとの位置関係を計算する
- カニさんがワナにかかったらシビレさせる
- カニさんがいるところが分岐点かどうかを判定する
- そのまま進む
- 反転させる

```

930 IF JX*JY<>0 AND RND(2)=1
GOTO 960
940 IF JX=1 AND SR$<>" " THE
N KD=3:GOTO 1060
950 IF JX=-1 AND SL$<>" " TH
EN KD=7:GOTO 1060
960 IF JY=1 AND SD$<>" " THE
N KD=5:GOTO 1060
970 IF JY=-1 AND SU$<>" " TH
EN KD=1:GOTO 1060
980 ' ■イケナイ
990 S=RND(4)+1
1000 ON S GOTO 1010,1020,103
0,1040
1010 IF SR$<>" " THEN KD=3 G
OTO 1060
1020 IF SL$<>" " THEN KD=7 G
OTO 1060
1030 IF SD$<>" " THEN KD=5 G
OTO 1060
1040 IF SU$<>" " THEN KD=1 G
OTO 1060
1050 GOTO 1010
1060 ' ■ウコ"ク
1070 DEF MOVE(F)=SPRITE (14,
KD,1,4,0,3):POSITION F,KX*8+
12,KY*8+16:MOVE F
1080 ' ■ワ"ツカッタ?
1090 IF CRASH(7)>=0 THEN PLA
Y "01B2AGFEDC":GOTO 1120
1100 ' ■
1110 GOTO 470
1120 ' ■オリ
1130 FOR JX=6 TO 8:READ S$:L
OCATE 9,JX:PRINT S$:NEXT
1140 SPRITE OFF
1150 DATA ■
1160 DATA ■ オシマイ ■
1170 DATA ■
1180 S=STRIG(0):IF S=0 GOTO
1180
1190 IF S=1 THEN RUN
1200 END

```

- 分岐点でペンペンに近付く方向をさがす
- 分岐点でペンペンに近付く方向に進めなかったら、ほかに進め方向をさがす
- いろいろ判定して決めた方向にカニさんを進ませる
- ペンペンがカニさんにつかまつたらゲームオーバーにする
- ● メインループの最初に戻る
- ゲームオーバーの表示をする
- STARTボタンが押されたら、またゲームを始める。それ以外のボタンでプログラムを終わる

めい ろ へん すう ペンペン迷路変数リスト

PS\$	ひろ 拾うものの文字列 もじれつ
PL	ひろ 拾うものの数 かず
PS	いくつまで拾ったか ひろ
C	ワナの数 かず
T	スコア
FW	ワナを置いたかどうかのフラグ お
W	ワナの BG キャラクタコード ピージー
F	どちらのカニさんを動かすかを決めるフラグ うご
KN	カニさんの数 かず
KX, KY	カニさんの座標 ざひょう
KD	カニさんの進む方向 すす はうこう
E()	カニさんの状態を示すフラグ じょうたい しめ
SR\$, SL\$, SU\$,	カニさんのまわりの迷路を知るための変数 かいろ し へんすう
SD\$, SK\$	

③。マリオワールド

A. 遊び方

プログラムを呼び出すときは、

F3キーを押すか、または
ゲーム [2,1] RETURN

と押してください。このゲームには、
BGグラフィックが付いています。

1人用ゲームですから、**I**コントローラを使います。ゲームが始まった

ときは、写真10のような画面になっています。床の上には0~9の10個の数字と10個のリンゴがあります。コントローラの中ボタンを使ってマリオを左右に歩かせたり、ハシゴを登り降りさせて、数字とリンゴを取ってください。数字は1個30点、リンゴは1個15点です。数字は0, 1, 2……9の順に取らないと、得点にはなりません。マリオは、床の端やハシゴから飛び降りることができます。コントローラのAボタンを押すと、マリオが真上にジャンプし、中ボタンの左か右と一緒に押すと、その方向にジャンプします。床の上に「=」がありますが、これはジャンプ台です。マリオがこの上に立ち止まるとき、自動的にひとつ上の床に飛び乗ることができます。数字を取るときや、ジャンプ台を使うときは、必ず立ち止まるようにしてください。立ち止まらないと、判定されません。

Aボタンを使ってうまくジャンプすると、二タニタを飛び越すこともできます。二タニタは画面の左右に消えると、またすぐ上方

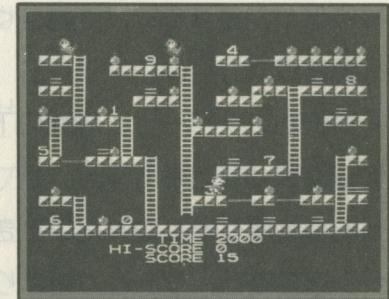


写真10

から出でてきます。

ハシゴを登りたいときは、ハシゴの真下で立ち止まってから、+ボタンを上に押しします。また、Aボタンを使いハシゴに向かってジャンプすると、すばやく登ることができます。



ゲームオーバーは、画面の下のTIMEが0になるか、二タニタにつかまつたときです。

もう一度ゲームをしたいときは、**I**コントローラのSTARTボタンを押すと、ゲームがはじまります。ゲームをやめたいときは、**STOP**キーを押してプログラムを止めてください。

B. 改造の方法

① BGグラフィックを変えてみましょう

マリオワールドは、BGグラフィックを変えて、自分の好きな画面にすることができます。STOPキーでプログラムを止めてから

ビージーツール リターン
B G T O O L R E T U R N

とキーを押して、BGグラフィック
モードにしてください。ハシゴや床
の場所を変えたり、リンゴの数を増
やしたりしてみましょう。ただし、
床やハシゴなどは決まったBGキャラ
ラクタを使ってください。それ以外のBGキャラクタは判定されま
せん。数字は0~9がそれぞれ1つずつあるようにしておきましょう。
(注) BGTOOL で BG グラフィックを改造してから、ゲームを始

(注) BGTOOL で BG グラフィックを改造してから、ゲームを始

めるときは、**E3** キーではなく、必ず **E8** キーを押してください。

かいぞう ピー ジー き
そういうないと 改造したBGグラフィックは消えてします。

いきに入ったBGグラフィックができたら、BGグラフィックモード
の「FILE」か、BASICのSAVES命令を使って、テープ
にセーブしておきましょう。テープからロードするときは、BGグ
ラフィックモードの「FILE」か、BASICのLOADS命令
を使います。

ないぞう
内蔵されたマリオワールドのプログラムと、テープにセーブしてあるBGグラフィックを組み合わせて遊ぶときは、まずF3キーを押すか。

ゲーム リターン
GAME RETURN

とキーを押して、プログラムを呼び出し、動き始めてから **STOP**
キーでゲームを止めてください。次にテープからBGグラフィック
をロードして、RUNしてください。なお、詳しくは、P.17 の
『2-3内蔵プログラムの使い方』を参照してください。

② BGグラフィックの色を変えてみましょう

リストの40行にPALETB命令があります。その後に新しく3つの行を追加します。

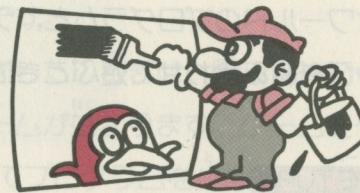
40 PALETB 3, 13, 48, 54, 22:... いかりくす以下略
パレット ピー リンゴ

41 PALETB 2, 13, 48, 54, 25
バレット ピー リンゴ

42 PALETB 1, 13, 48, 54, 48
パレット ピー
ゆか ゆか あいだ
床と床の間のバー

43 PALETB \varnothing , 4, 48, 54, 39
パック 文字とハシゴ 床とジャンプ台

それぞれの色を自由に変えることができます。好きな色にしたいときは、色コード表（P. 32）に従って変更してください。文字とハシゴの色を黒にすると、後でリストを出したときに、見えなくなってしまいますから注意しましょう。もしそうなってしまったときは、**CTR**キーを押しながら**D**キーを押せば、元の色（白）に戻ります。



③ アニメキャラクタを変えてみましょう

280行の数字を変えると、ニタニタがほかのアニメキャラクタに変わります。

280 デザイン ムーブ スプライト
DEF MOVE (1) = SPRITE (11, D, ...
11はニタニタのことですから、0~15のほかの数字にすればよいのです。数字をいくつにすると、どのアニメキャラクタになるのかは、本書の付録P.128（下段）をご覧ください。

540行と630行の数字を変更すると、ハシゴの登り降りや左右に動くときのアニメキャラクタが変わります。

540 デザイン ムーブ スプライト
DEF MOVE (7) = SPRITE (0, V, 1 ...
630 イフ ケン THEN DEF MOVE (7) =
スプライト SPRITE (\varnothing , V, 4-K,以下略

どちらの行も \varnothing のマリオになっていますが、たとえば540行は4

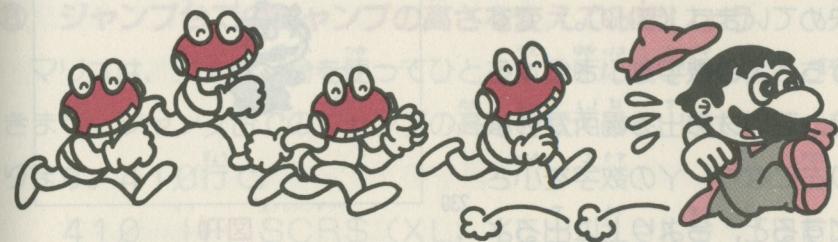
に、630行は13に、というように変えます。

④ ニタニタの数を変えてみましょう

ニタニタは2匹で追いかけますが、たった1カ所変えるだけで、増やしたり減らしたりすることができます。変更するのは140行で、

140 T=800:W= \varnothing :NN=1:X=116.....以下略

この「1」を、ほかの数字にすればよいのです。ただし、この数字はニタニタの数より、1だけ小さい数にしなければいけません。今はニタニタが2匹なので、140行の数字は、2より1少ない数字になっているわけです。4匹（数字は3）以上になると、マリオがときどき見えなくなります。



⑤ 得点を変えてみましょう

数字は1個30点、リンゴは1個15点となっていますが、この得点は変えることができます。たとえば、数字を1個100点にするときは、

420 イフ アスキースクリーングラード
THEN LOCATE XL, YL-1:PRINT "
":Z=Z+1:W=W+30:.....以下略

の30を100に変えてください。

リンゴの得点を1個50点にするときは、

880 LOCATE XB, YL-1 : PRINT " "
: R=R+1 : W=W+15.....以下略

15を50に変えてください。数字とリンゴの得点の合計が、32767を越えるとエラーになります。

⑥ マリオが最初に出てくる場所を変えてみましょう

マリオが画面に出てくる場所は140行のXとYで決めています。

140 T=800:W=0:NN=1:X=116:Y=1
27:POSITION.....以下略

Xは、画面の左端からの位置、

Yは、画面の上からの位置を

決めています(図10)。です

から、Xの数字を小さくする

と、マリオの出る場所が今よ

り左になります。Yの数字を小さ

くすると、今より上に出るよ

うになります。この数字を変えるときは、Xが16~224、Yが

16~168くらいにしておきましょう。BGグラフィックで自分

の画面を作った人は、その画面に合わせて、マリオを出す場所を変

えた方がよいでしょう。

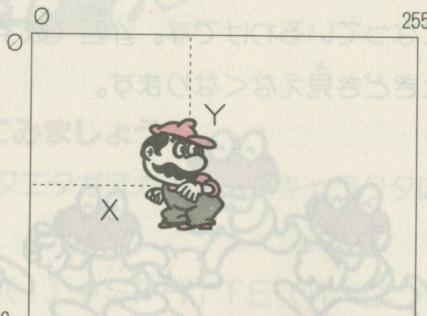


図10

⑦ タイマーを付けてゲーム時間を変えてみましょう

画面の下にTIMEという表示があります。この数字が800から100ずつ減っていって、0でゲームオーバーになっていますが、

800という数字を変えることによって長くしたり短くしたりもできます。リストの最初のところに、

140 T=800:W=0:.....以下略

ゲーム時間を決める数字がありますから、その数字を変えてください。ただし、BGグラフィックで、リンゴの数が15個以内で時間だけを長くすると、数字とリンゴを全部取った後、時間があり余つて、ただマリオを動かすだけになってしまいます。リンゴを16個以上にしておけば、数字とリンゴを全部取ると、また新しく出てきますから、時間を長くしてもゲームを続けることができます。ですから、元のBGグラフィック画面を使って遊ぶときは、時間を短くした方がよいでしょう。

⑧ ジャンプ台でのジャンプの高さを変えてみましょう

マリオは、ジャンプ台を使ってひとつ上の床に飛び乗ることができます。ジャンプ台でのジャンプの高さを変えると、おもしろくなります。410行で、

410 IF SCR\$ (XL, YL-1) = K\$ THEN
G=-15:GOTO 770

G=-15の数字が、ジャンプの高さを決めています。

この数字を-20に変えると、高くジャンプするようになります。

ただし、BGグラフィックの作り方によって、ものすごく高く飛んだり、すぐ上の床に飛び乗るだけだつたりします。BGグラフィックを作るときには、画面のあまり上のほうに、ジャンプ台を置かないようにしましょう。ジャンプしたときに、アニメキャラクタが画面からはみ出てエラーになり、プログラムが終わってしまうことが

あります。

次は、逆に低くしてみましょう。G=-15の数字を、-10になると、もう上の床に飛び上がれなくなってしまいます。

さて、最後に少し愉快なプログラムにしましょう。G=-3にしてください。これでマリオがジャンプ台のところで立ち止まると、ダンシングマリオになってしまいます。

今度は、G=0にしてRUNしてみてください。ジャンプ台が落とし穴に変わってしまいます。ジャンプ台のところでマリオが立ち止まると、ひとつ下の床まで落ちます。これはゲームのルールが変わったことになると、また楽しく遊べます。落とし穴にしたときは、いちばん下の床から落ちると、ゲームオーバーになります。



⑨ Aボタンでのジャンプ幅を変えてみましょう

ニタニタが近づいてきたときに、Aボタンを押してうまくジャンプすると、マリオはニタニタを、飛び越すことができますが、改造すると、飛び越せなくなったり、もっと楽に飛び越すことができるようになります。



```
770 F=1:SPRITE S, X, Y:ERA 7::  
SPRITE 3:POSITION 7, X, Y:S2=S  
:K=K* ((S=2) .....以下略
```

リストの770行に、「K=K*」というところがありますが、これを変更すると、ジャンプ幅を変えることができます。たとえば、

$K = (K - 1) *$

になると、ぎりぎりのところでニタニタを飛び越せなくなります。

$K = (K - 3) *$

になると、+ボタンを押しながらジャンプしても、ほとんど前へ進まなくなってしまいます。

$K = (K + 2) *$

のようにすると、ニタニタの飛び越しが、少し楽になります。

ゲームとしては使えませんが、たとえば、スーパージャンプにしたいときは、

$K = (K + 10) *$

になると、ニタニタを追いかけて、飛び越すこともできるようになります。それから、

$K = (K - 20) *$

になると、マリオは後ろに飛んでしまいます。



⑩ Aボタンでのジャンプ力(跳跃力)を変えてみましょう

⑨では、マリオがジャンプしたときに、どのくらい遠くまで飛べるのか、いろいろ改造してみました。そこで今度は、どのくらい高くジャンプできるのか、Aボタンのジャンプ力を、いろいろ変えて

みましょう。

改造する行は、760行です。

760 G=-9 : X=X+.....以下略

760行のはじめのところに、「G=-9」というのがあります。この「-9」という数字を、「-15」や「-20」のような、マイナスのついた大きな数字になると、より高くジャンプできるようになります。

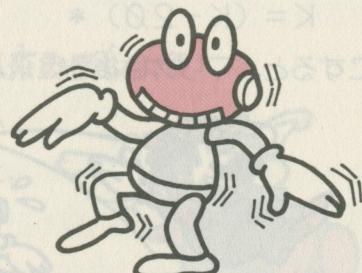
ところが、あまり上方からジャンプすると、画面の外に飛び出してしまい、「OK」の文字が表示され、プログラムが止まってしまいます。そこで、次の1行を新しく追加してください。

1025 IF Y<16 THEN Y=16:YL=2:
RESUME

この1行を追加しておけば、画面の上方でジャンプしても、プログラムの実行が、中止されることはありません。

なお、ジャンプする高さが高くなるほど、1回のジャンプにかかる時間が長くなるために、二タニタの動きもその長さに合わせて、少しきちなくなることがあります。また、画面の左右両端でジャンプすると、マリオが空中で止まってしまい、キーを押しても動かなくなってしまうことがあります。もし、止まってしまったら、STOPキーを押して、一度プログラムを中止させ、RUNし直してください。

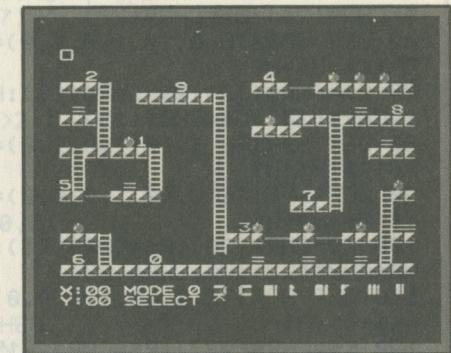
ジャンプする高さを高くしすぎると、たとえば「-25」や「-



30」という大きな数字にすると、ジャンプした後、床を通り越して、落ちてしまうことがあります。ですから、数字は「-20」くらいまでにしておいた方がよいでしょう。



ジャンプの高さを変えたときは、写真IIのように、BGTOOL命令を使って、ハシゴや床を少し消しておきます。そして、広い場所を作つておいて、そこでジャンプすると、改造の効果をよく見ることができますから、試してみてください。



写真II

⑪ マリオの歩き始めをすばやくしましよう

マリオを左右に動かそうとするとき、中ボタンの左か右を押すと、少しの間ゆっくりと歩いてから、速く歩きだしています。これは、動きが自然に見えるような工夫なのですが、すばやく動かしたい人は、630行を変えてください。

630 IF K<4 THEN DEF MOVE (7)=
SPRITE (0, V, 4-K, 255) :.....以下略

()の中の4-Kというところを、数字の1にします。すると、マリオはいきなり速く歩き始めるようになります。

マリオワールドプログラムリスト

```

10 CGSET 1,0
20 ON ERROR GOTO 910
30 W=F=K=T=R=Z=NI=NN=XE=YE=X
N=YN=XB=XL=YL=S2=S1=S=G=V=X=
Y=I=D:H=0:S$=G$
40 PALETB 3,13,48,54,22:SPRI
TE ON
50 K$=CHR$(24):R$=CHR$(25):H
$=CHR$(26):U$=CHR$(27)
60 DEF SPRITE 0,(0,1,0,1,0)=
R$+K$+U$+H$
70 K$=CHR$(12):R$=CHR$(13):H
$=CHR$(14):U$=CHR$(15)
80 DEF SPRITE 1,(0,1,0,1,0)=
R$+K$+H$+U$
90 DEF SPRITE 2,(0,1,0,0,0)=
K$+R$+H$+U$
100 K$=CHR$(20):R$=CHR$(21):
H$=CHR$(22):U$=CHR$(23)
110 DEF SPRITE 3,(0,1,0,0,0)=
=K$+R$+H$+U$
120 K$=CHR$(227):R$=CHR$(215
):H$=CHR$(210):U$=CHR$(192):
DIM Q(3),D(3)
130 'ショキ セッティ
140 T=800:W=0:NN=1:X=116:Y=1
27:POSITION 7,X,Y:GOSUB 890:
V=0:GOTO 540
150 'タイム トトクン ノ ハンテイ
160 T=T-1:IF T MOD 10=0 THEN
LOCATE 14,21:PRINT T;"":I
F T=0 THEN 660
170 IF Z=58 THEN IF R>15 THE
N GOSUB 890
180 'ニタニタ ノ フログ"ラム
190 I=-(I+1)*(I<>NN)
200 XN=XPOS(I):YN=YPOS(I)
210 IF XN<12 OR XN>222 OR YN
>188 THEN XN=RND(27)*8+12:YN
=7:POSITION I,XN,YN:D=0:Q(I)
=5:GOTO 280
220 XE=(XN-8)/8:YE=(YN-2)/8
230 G$=SCR$(XE,YE)

```

●準備

●ジャンプしているときの
マリオのパターンを決める

●準備

●時間とスコアの判定

72 ※本書リストに出てくる「＊」は、「＊」と同じ文字です。画面に出てくる文字
と、多少異なりますので、ご了承ください。

```

240 D=Q(I)
250 IF G$<>U$ THEN IF YL<>YE
OR G$<>H$ THEN D=5:Q(I)=3-4
*(X<XN):GOTO 270
260 IF D(I)=D THEN 300
270 XN=XE*8+12:YN=YE*8+7:POS
ITION I,XN,YN
280 DEF MOVE(I)=SPRITE(11,D,
2,255+251*(D=5)):MOVE I:D(I)
=D
290 'フ"ツカリ ノ ハンテイ
300 NI=CRASH(7):IF NI>-1 THE
N IF ABS(YPOS(NI)-YPOS(7))<1
4 AND ABS(XPOS(NI)-XPOS(7))<
14 THEN 660
310 IF F=1 THEN 780
320 'マリオ ノ フログ"ラム
330 S1=S:S=STICK(0):IF S>2 T
HEN S=3-(S>7)
340 X=XPOS(7):Y=YPOS(7)
350 XL=X/8-1:YL=(Y-2)/8
360 ON S GOTO 550,550,480,51
0
370 '
380 V=0:K=0:CUT 7
390 IF SCR$(XL,YL)=-" " THEN
IF SCR$(XL,YL-1)<>H$ THEN GO
SUB 870:GOTO 750
400 XB=XL+(S1=1)-(S1=2):IF S
CR$(XB,YL-1)=R$ THEN GOSUB 8
80
410 IF SCR$(XL,YL-1)=K$ THEN
G=-15:GOTO 770
420 IF ASC(SCR$(XL,YL-1))=Z
THEN LOCATE XL,YL-1:PRINT " "
:Z=Z+1:W=W+30:LOCATE 14,23:
PRINT W;
430 IF STRIG(0)<>8 THEN 470
440 IF SCR$(XL-1,YL)=U$ THEN
760
450 IF SCR$(XL,YL)=U$ THEN 7
60
460 IF SCR$(XL+1,YL)=U$ THEN
760
470 GOTO 160
480 V=5:G$=SCR$(XL,YL)
490 IF G$<>H$ THEN IF G$<>U$
THEN S=0:V=0:GOTO 750

```

●ニタニタがマリオを追いかける

●マリオがニタニタにつかまつた
はんてい
判定

●①コントローラの中央ボタンが
お押された判定

●数字をひろったら30点

●マリオの足元の床の判定

```

500 GOTO 520
510 V=1:G$=SCR$(XL,YL-1)
520 IF G$<>H$ THEN S=0:GOTO
380
530 IF S1=0 OR K>0 THEN GOSU
B 870:K=0
540 DEF MOVE(7)=SPRITE(0,V,1
,4):MOVE 7:SPRITE S2:PAUSE 5
:GOTO 160
550 V=4*S-1:K=1-(K+(K>3))*(S
=S1)
560 G$=SCR$(XL+1+2*(S=2),YL)
:S$=SCR$(XL,YL)
570 XB=XL+1+2*(S=1):IF K>2 T
HEN IF SCR$(XB,YL-1)=R$ THEN
GOSUB 880
580 IF SCR$(XL,YL-1)=R$ THEN
XB=XL:GOSUB 880
590 IF G$<>U$ AND G$<>H$ THE
N X=X+6*(S$<>" ")*(S=1)-(S=
2)):GOTO 750
600 IF S$=" " THEN S=S1:GOTO
750
610 IF STRIG(0)=8 THEN IF G$=
U$ OR S$=U$ OR SCR$(XB,YL)=
U$ THEN 760
620 NI=CRASH(7):IF NI>-1 THE
N IF ABS(XPOS(NI)-XPOS(7))<1
4 THEN 660
630 IF K<4 THEN DEF MOVE(7)=
SPRITE(0,V,4-K,255):MOVE 7:S
PRITE S2
640 GOTO 160
650 'オクリ ノ フログ'ラム
660 SPRITE S:SPRITE S1:SPRIT
E S2:ERA 7
670 FOR I=0 TO NN:ERA I:POSI
TION I,0,0:D(I)=0:NEXT
680 LOCATE 8,9:PRINT "*****"
*****"
690 LOCATE 8,10:PRINT "* GAME
OVER *"
700 LOCATE 8,11:PRINT "*****"
*****"
710 IF W>H THEN H=W:LOCATE 1
4,22:PRINT H;
720 IF STRIG(0)=1 THEN 140
730 GOTO 720

```

●マリオがハシゴを登り降りする

●マリオを左右に歩かせて
リンゴや床の判定をする

●ゲームオーバーを表示して
再ゲームの入力待ち

```

740 'シ"ヤン" ノ フログ'ラム
750 G=2:Y=Y+3:GOTO 770
760 G=-9:X=X+(2*K+6)*((S=2)-
(S=1)):Y=Y-12
770 F=1:SPRITE S,X,Y:ERA 7:::
SPRITE 3:POSITION 7,X,Y:S2=S
:K=K*((S=2)-(S=1)):GOTO 160
780 NI=CRASH(7):IF NI>-1 THE
N IF ABS(YPOS(NI))-Y<12 THEN
660
790 G=G+3+3*(G>6):X=X+K:Y=Y+
G:SPRITE S,X,Y:POSITION 7,X,
Y
800 XL=X/8-1:YL=(Y-2)/8:G$=S
CR$(XL,YL):S$=SCR$(XL,YL-1)
810 IF G$=U$ THEN GOSUB 860:
K=1:GOTO 540
820 IF S$=H$ THEN GOSUB 860:
SPRITE 3,X,Y:SPRITE S2:S2=3:
GOTO 160
830 IF G$=H$ THEN GOSUB 860:
V=0:GOTO 540
840 GOTO 780
850 'SUB ノ フログ'ラム
860 GOSUB 870:F=0:K=0:SPRITE
S,X,Y:RETURN
870 CUT 7:X=XL*8+12:Y=YL*8+7
:POSITION 7,X,Y:RETURN
880 LOCATE XB,YL-1:PRINT " "
:R=R+1:W=W+15:LOCATE 14,23:P
RINT W:;RETURN
890 VIEW:I=0:F=0:K=0:C=0:R=0
:Z=48:LOCATE 10,21:PRINT "TIM
E";T:LOCATE 6,22:PRINT "HI-SC
ORE";H:LOCATE 9,23:PRINT "SCO
RE";W;
900 RETURN
910 'エラー シヨリ ノ フログ'ラム
920 IF ERL=560 THEN XL=-27*(X
L<1):GOSUB 870:MOVE 7:RESUM
E 630
930 IF ERL=390 THEN XL=-27*(X
L<0):GOSUB 870:S=S1:RESUME
550
940 IF YL>23 THEN RESUME 660
950 IF XL>27 THEN XL=0:GOSUB
870:RESUME

```

●ジャンプのプログラム

●いろいろなサブルーチン

```

960 IF XL<0 THEN XL=27:GOSUB
870:RESUME
970 IF XB>27 THEN XB=0:MOVE7
:RESUME
980 IF XB<0 THEN XB=27:MOVE7
:RESUME
990 IF XL+1>27 THEN XL=0:GOSUB
870:RESUME
1000 IF XL-1<0 THEN XL=27:GOSUB
870:RESUME
1010 IF X<12 THEN X=228:RESUME
1020 IF X>228 THEN X=12:RESUME
1030 END

```

● エラーが出たときの処理

マリオワールド変数リスト

X, Y	マリオのスプライト座標 X, Y, YL	R	リンゴを拾った数 次に拾う数字のコード
XN, YN	ニタニタのスプライト座標 XE, YE	K\$	ジャンプ台のBGキャラクタ
NN	ニタニタの数 - 1	R\$H	リンゴのBGキャラクタ
NI	マリオが衝突したニタニタの番号	H\$	ハシゴのBGキャラクタ
DC()	ニタニタが動いた方向	U\$	床のBGキャラクタ
QC()	ニタニタが次に動く方向	S	マリオの方向、スプライト番号
I	ニタニタのMOVE番号	V	マリオが動く方向
W	ニタニタの得点	K	マリオの速度、ジャンプの飛距離
T	制限時間	G	ジャンプの下向きの速度
		F	ジャンプ中のフラグ

4. スターキラー

A. 遊び方

プログラムを呼び出すときは、

F4キーを押すか、または

ゲーム リターン
GAME **③[RETURN]**

とキーを押してください。

プログラムが始まると、**写真**

12のような画面になります。こ

のゲームは、1人用と2人用が

選べるようになっていますから、

1人で遊ぶときは、キーボードの**N**キーを押し、2人で遊ぶときは、

Yキーを押してください。1人で遊ぶときは、相手の宇宙船をコン

ピュータが操作します。

1人用ゲームか2人用ゲームかを決めると、ゲームが始まります。

プレーヤーの動かす宇宙船は、ゲームが始まつても動き出しません。

コントローラの中ボタンを押してください。これで宇宙船が、それ

ぞの陣地から動き出しますので、コントロールしてください。

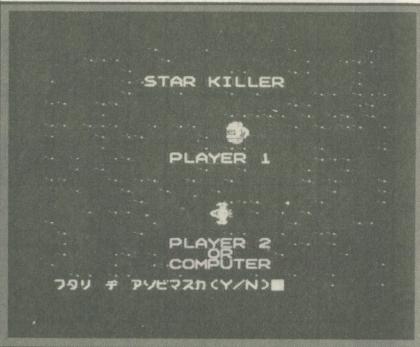
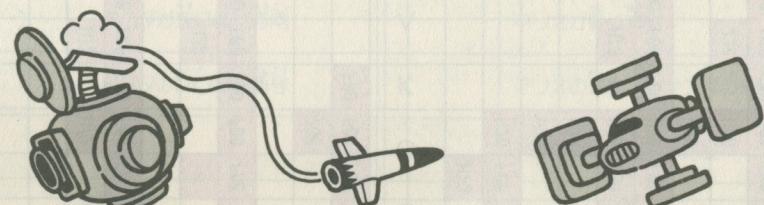
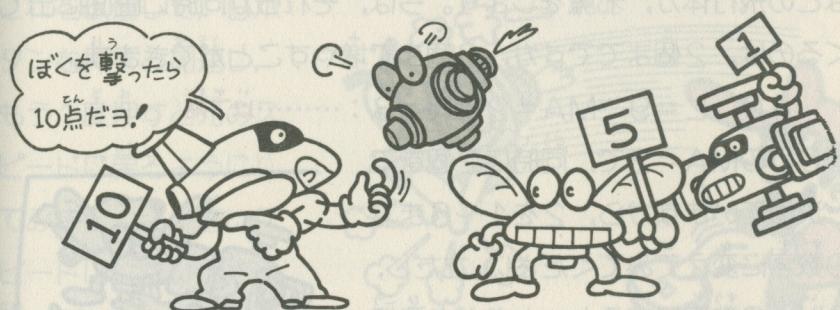


写真12

Aボタンを押すと、ミサイルが発射されます。このミサイルで相手の宇宙船や、そのほかの飛行体を撃ってください。相手の宇宙船に当たると10点、ファイターフライに当たると5点、そのほかの飛行体に当たると1点です。

Bボタンは、ブレーキです。Bボタンを押すと、宇宙船のスピードが遅くなりますが、そのかわり、こまわりがきくようになります。宇宙船のスピードを元に戻したいときは、もう1度Bボタンを押してください。

自分の宇宙船の数は、初めは4隻です。スピナーやファイアーボールなどの飛行体が、3個ずつ次々と現れ、ひと巡りすると、またスピナーに戻りますが、そのときに自分の宇宙船が、1隻増えます。ですから、得点の高い相手の宇宙船ばかりではなく、そのほかのキャラクタも、どんどんと撃ってください。



1人用ゲームのときは、自分の宇宙船がなくなつたときに、ゲームオーバーになります。2人用ゲームのときは、2人とも宇宙船がなくなつたときに、ゲームオーバーとなります。2人用ゲームで、1方の宇宙船がなくなつたときには、もう1方の宇宙船がなくなつてしまふまで、コンピュータが相手をします。

B. 改造の方法

① 最初に持っている宇宙船の数を変えてみましょう

1人用ゲームでも2人用ゲームでも、プレーヤーの持っている宇宙船の数は、ゲームが始まったときには4隻ずつです。この数を変えてみましょう。2人用ゲームのときは、それぞれの宇宙船の数を変えて、ハンディをつけることができます。

40 L=9:MA=2:N=3:V=7:C=7:S(0)
=4:S(1)=4
プレーヤー1の宇宙船の数 プレイヤー2の宇宙船の数

② 出てくる飛行体の数を変えてみましょう

宇宙船をコントロールしていると、スピナーやファイターフライなどの飛行体が、邪魔をします。今は、それらが同時に画面に出てくるのは、2個までですが、6個まで増やすことができます。

40 L=9:MA=2:N=3:……以下略
40行のMA=2で、同時に2個までと決めていますから、2を1~6まで数字に変えてみてください。ただし、3以上の数字になると、ときどき飛行体が見えなくなることがあります。7以上の数になるとエラーになり、また、0にしても1個は必ず出るようになっています。



飛行体は、スピナーで始まって、3個命中させると、次の飛行体に変わるようにになっています。

この数を変えてみましょう。

40 L=9:MA=2:N=3:V=7:……以下略

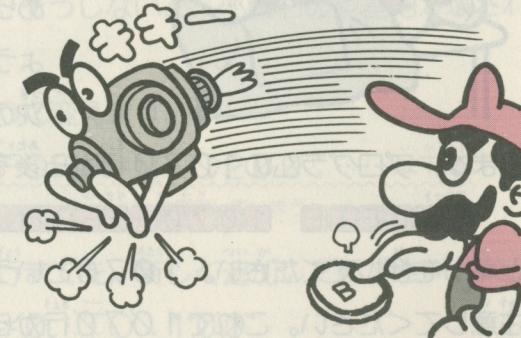
今はN=3になっています。この数字を1以上の適当な数に変えてみましょう。大きな数にすると、すべての飛行体が現れた後、元のスピナーに戻るまでの時間が、長くなります。そうすると、宇宙船が、1隻増えるのも遅くなります。これは、ゲームが難しくなったということです。

③ 宇宙船のスピードを変えてみましょう

今のBボタンは、ブレーキボタンですから、押すと宇宙船のスピードが遅くなっています。もう一度押すと、元に戻るようになっています。これを改造して、Bボタンを押すと加速し、もう一度押すと元のスピードに戻るようにしてみましょう。速いスピードにしたいときに、試してみてください。

40 L=9:MA=2:N=3:V=7:C=7:……以下略

このV=7の数字を3にすると、Bボタンが加速ボタンに変わります。ただし、この改造をすると、Bボタンを押して加速したときに、宇宙船の速さが、ミサイルと同じになってしまいます。加速中にミサイルを発射すると、ミサイルを抱えて飛んでいるようになります。



宇宙船の方向を変えると、ミサイルだけが機体から切り離されて、元の進行方向に飛んでいくことになります。命中の判定などは、前と同じですから、攻撃の仕方を変えて楽しんでください。

数字を3ではなく、たとえば10くらいになると、Bボタンは急ブレーキボタンになります。

④ 1人用・2人用を選ぶ画面をなくしてみましょう



このプログラムは、RUNすると必ず、1人用にするか2人用にするかの入力があるまで止まっています。1人で何回もプレイするときには、RUNするとすぐにゲームが始まられるように、改造しましよう。これは大手術ですが、次のようにしてください。

まず、プログラムの1070行から後を、全部消してしまいます。
DELETE 1070—RETURN

とキーを押してください。1070という数字を間違えないように注意してください。これで1070行から最後の行までが、まとめて消えました。

では、90行を次のように変えてください。

```
90 GOSUB 1070
90 IF STRIG(0) = 0 THEN 90
```

こうしておくと、RUNした後、コントローラのA, B, START,

セレクトSELECTのどのボタンを押しても、すぐにゲームが始まるようになります。ゲームオーバーになったときも、どれかを押せばすぐにゲームが始まります。

最後に、新しく次の1行を追加してください。この行は、1人用にするか、2人用にするかで少し異なります。

(1人用) 85 M=1 (2人用) 85 M=-1

この数字を1にしておけば、1人用になります。

【注意】 ここから後の、「邪魔ものをおいてみましょう」と「基地を作つてみましょう」を行なう人は、必ずプログラムの一部を、次のように改造してください。そうしないと、邪魔ものや基地が壊されてしまいます。

まず100行を次のように変えます。

```
{ 100 VIEW :FOR P=0 TO 200: .....以下略
 100 VIEW
```

次に、101行～104行までを、新しく追加してください。

```
101 FOR P=0 TO 100
102 X=RND(28):Y=RND(23):IF SCR$(X, Y) <> " " THEN 102
103 LOCATE X, Y:PRINT CHR$(205);
104 NEXT:P=0
```



⑤ 邪魔ものをおいてみましょう

このプログラムでは、宇宙船の飛び回る宇宙空間に、邪魔ものをおくことができます。宇宙船が邪魔ものにぶつかると、爆発してしまいますから、ゲームがたいへん難しくなります。邪魔ものをおくときには、2つのルールがあります。次の説明をよく読んでから、試してください。

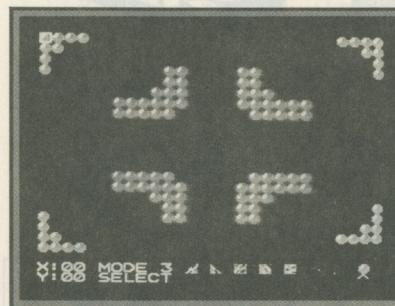


写真13

な画面になります。このとき、「へ」マークがある「●」が、邪魔ものに使えるBGキャラクタです。これを使わないと、邪魔ものと判定してもらえません。これがルールその1です。

次に、このBGキャラクタをおくときの配色番号は、このままの「0」か、

RETURNキーを3回押したときの「3」にしなくてはいけません。「0」が「3」以外の配色番号になると、やはり邪魔

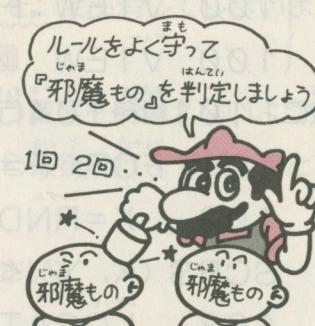
ものと判定してもらえません。これがルールその2です。

この2つのルールを守れば、画面上のどこにおくかは自由です。
(ただし、宇宙船のスタート地点に邪魔ものをおくと、ゲームを始め

まず、STOPキーを押して、

プログラムを止めて、
ビージーツール リターン
BG TOOL RETURN

とキーを押して、BGグラフィックモードにしてから、CLR HOMEキーを6回、DELキーを7回押してください。これで、写真13のよう



ることができます。うまくデザインすると、ゲームがとてもおもしろくなりますから、いろいろと試してください。

邪魔もののおき方によっては、宇宙船の操縦が、とても難しくなります。でも、邪魔ものは、ミサイルで消すことができます。得点にはなりませんが、ゲームが始まつたら、邪魔ものをどんどん撃つていくのも、ひとつの作戦になります。ただし、宇宙船以外のアニメキャラクタは、邪魔ものにぶつかっても爆発しません。

⑥ 基地を作つてみましょう

それぞれ自分専用の基地を作りましょう。基地を作るときも、BG TOOL命令で、BGグラフィックにします。この場合も、2つのルールがあります。BGグラフィックモードにする方法と、ルールその1は、「邪魔ものをおいてみましょう」と同じです。



ルールその2は、プレーヤー1とプレーヤー2の基地を作るときの配色番号です。

プレーヤー1の基地………配色番号 1 で作る

プレーヤー2の基地………配色番号 2 で作る

と決まっていますから、配色番号を間違えないようにしてください。
基地のかたちや、どこに作るかは自由にできます。

それぞれの基地を作ると、

配色番号1の基地……プレーヤー2がぶつかると爆発
 配色番号2の基地……プレーヤー1がぶつかると爆発
 のようになりますから、危なくなったら自分の基地に逃げ込みます。
 このとき、Bボタンの急ブレーキを使って作戦に変化をつけてください。

ミサイルの攻撃を受けると基地はこわれてしまいます。基地と邪魔のものを組み合わせておけば、さらに楽しさが広がります。

スターキラー変数リスト

T	エストリガー STRIG()の値	V	Bボタンを押したときの速度
S	スティック STICK()の値	X0, Y0	プレーヤー1のホームポジション
U	表示されている飛行体の数	X1, Y1	プレーヤー2のホームポジション
N	やつづける全飛行体数	D()	宇宙船が動く方向
C	飛行体のアニメキャラクタ番号	V()	宇宙船の速度
MA	一画面に表示する飛行体の数	F(0) ~F(7)	何と何がぶつかったかのフラグ
L	コンピュータの強さ	F(8)	ファイターフライの出現フラグ
M	1人ゲームか2人ゲームか	SC()	スコア
P	0: Iコントローラ 1: IIコントローラ	O	攻撃したMOVE番号など
	1: IIコントローラ	D	攻撃されたMOVE番号など

スタークリープログラムリスト

```

10 ''AREA 88 V7.3"
20 T=0=U=C=H=D=I=K=L=M=N=P=S
=V=X=Y=MA=X0=X1=Y0=Y1=PP=Q
30 DIM D(1),V(1),F(8),SC(1),
S(1)
40 L=9:MA=2:N=3:V=7:C=7:S(0)
=4:S(1)=4
50 X0=70:Y0=80:X1=110:Y1=60
60 D(0)=3:V(0)=2:D(1)=7:V(1)
=2:F(0)=3:F(1)=3:MA=MA+1:U=N
:M=1
70 SPRITE ON:CGSET 1,1:PLAY*
T1Y2M1V903R1":PALETS 1,13,48
,38,18
80 FOR I=15 TO 18:A$=A$+SCR$
(I,22):NEXT:H=VAL(A$)
90 GOSUB 1070
100 VIEW:FORP=0TO200:LOCATER
ND(27),RND(23):PRINTCHR$(205
);:NEXT:P=0
110
120 I=P+6:IF F(P) THEN ON F(
P) GOSUB 670,700,680:GOTO 15
0
130 IF M=P THEN GOSUB 390:GO
TO 150
140 GOSUB 210
150 IF MOVE(I)=0 THEN ERA I
160 GOSUB 530
170 GOSUB 800
180 ON F(8)+1 GOSUB 940,1040
190 P=1-P:GOTO 120
200
210 S=STICK(P):T=STRIG(P)
220 IF T<>4 THEN F(I)=1
230 IF S+T=0 THEN RETURN
240 IFS=0THEN350
250 S=-1*(S=8)-3*(S=1)-5*(S=
4)-7*(S=2)-2*(S=9)-4*(S=5)-6
*(S=6)-8*(S=10)
260 Q=D(P)
270 IF S=Q THEN 350
280 IF Q>4 THEN 310

```

●準備

ひと り よう ふた り よう そな
● 1人用か2人用を選ぶサブルー

→ チンを呼ぶ

● BGを呼び出して星を描く

●メインルーチン

※本書リストに出てくる「*」は、「*」と同じ文字です。画面に出てくる文字
 と、多少異なりますので、ご了承ください。

```

290 IF (S>Q AND S<=Q+4) THEN
320 GOTO 330
310 IF (S)=Q-4 AND S<(Q)THEN
330
320 Q=Q+1:GOTO 340
330 Q=Q-1
340 PP=-1*(Q>8)-8*(Q<1):Q=-Q
*(PP=Q)+PP:D(P)=Q
350 IF T AND4 THEN IF F(I) T
HEN V(P)=V-V(P):F(I)=0
360 DEF MOVE(P)=SPRITE (P+8,
D(P),V(P),255,1):MOVE P
370 IF T>7 THEN DEF MOVE(I)=
SPRITE(12,D(P),1,80,0,P):POS
ITION I,(XPOS(P)+4)MOD256,(Y
POS(P)+4)MOD256:MOVE I
380 RETURN
390
400 K=K-1:IF K>0 THEN RETURN
410 T=0:D=0:X=(XPOS(1-P)-XPO
S(P))/13:Y=(YPOS(1-P)-YPOS(P
))/13
420 IF X>0 THEN D=3
430 IF X<0 THEN D=7
440 IF Y>0 THEN D=D+5
450 IF Y<0 THEN D=D+1:IF X<0
THEN D=16
460 IF X AND Y THEN D=D/2
470 S=2:0=D-D(P):IF(0<0 AND
0<5)OR 0<-3 THEN S=1
480 IF 0=0 THEN S=0
490 IF V(1-P)<>V(P) THEN T=4
500 IF MOVE(I)=0 OR RND(10)=
0 THEN T=T+8
510 IF X*X+Y*Y>200 THEN RETU
RN
520 K=RND(L):GOTO 220
530
540 D=CRASH(0):IF D>0 AND D<
6 THEN 0=0:GOSUB 590
550 D=CRASH(1):IF D>1 AND D<
6 THEN 0=1:GOSUB 590
560 D=CRASH(6):IF D>0 AND D<
6 THEN 0=6:GOSUB 610
570 D=CRASH(7):IF D=0 OR (D>

```

モード

- プボタンやトリガボタンが押された判定

● 1人用のとき相手の宇宙船を動かす

動き

1 AND D<6)THEN 0=7:GOSUB 610

580 RETURN

```

590 X=ABS(XPOS(0)-XPOS(D)):Y
=ABS(YPOS(0)-YPOS(D)):IF X<1
0 AND Y<10 THEN IF MOVE(0) A
ND MOVE(D) AND F(D)=0 AND F(
0)=0 THEN F(0)=1:F(D)=1:IF D
>1 THEN 650
600 RETURN
610 X=ABS(XPOS(0)-XPOS(D)-4)
:Y=ABS(YPOS(0)-YPOS(D)-4):IF
X<8 AND Y<8 THEN IF MOVE(0)
AND MOVE(D) AND F(D)=0 THEN
ERA 0:F(D)=1:GOTO 630
620 RETURN
630 IF D<2 THEN SC(0-6)=SC(0
-6)+10:RETURN
640 SC(0-6)=SC(0-6)+1+F(8)*4

```

```

650 DEF MOVE(D)=SPRITE(10,RN
D(9),RND(3)+4,10):MOVE D:PLA
Y"04C":RETURN
660

```

```

670 F(P)=2:DEF MOVE(P)=SPRIT
E(10,D(P),V(P),70/V(P),0):MO
VE P:PLAY "O"+HEX$(P+3)+"GG":
S(P)=S(P)-1:RETURN
680 IF P=M OR STRIG(P)+STICK
(P) THEN F(P)=0:POSITION P,X
0+P*X1,Y0+P*Y1
690 RETURN

```

```

700 IF MOVE(P) THEN RETURN
710 F(P)=3:ERA P
720 IF SC(0)>H AND M<>0 THEN
H=SC(0)
730 IF SC(1)>H AND M<>1 THEN
H=SC(1)
740 IF L<1 THEN L=5

```

```

750 LOCATE 0,20:PRINT "SCORE
",SC(0),SC(1),"LEVEL":PRINT
"SHIPS",S(0),S(1),L;"":PRINT
"HI-SCORE",H,
760 IF S(P)>0 THEN RETURN
770 IF M=P THEN L=L-1:S(P)=4
:GOTO 740
780 IF M=-1 THEN M=P:PRINT P
;" COM":PLAY "EE":RETURN

```

- どのアニメキャラクタ同志がぶつかったかなどの細かい判定をする

● 宇宙船を爆発させる

- 宇宙船が消えた後、コントローラのボタンが押されたら、次の宇宙船を出す

● ゲームオーバーの表示をして再ゲーム

```

790 LOCATE 10,10:PRINT "GAME
OVER":PLAY"01C5CDED6C4C5":P
AUSE,200:RUN
800
810 X=XPOS(P)/8-1:Y=YPOS(P)/
8-2
820 IF X<0 OR X>27 OR Y<0 OR
Y>23 THEN 880
830 IF SCR$(X,Y)<>"•"THEN 88
840 S=ASC(SCR$(X,Y,1))
850 IF S=P+1 THEN 880
860 IF F(P)=0 THEN F(P)=1
870
880 IF MOVE(I)=0 THEN RETURN
890 X=(XPOS(I)-12)/8:Y=(YPOS
(I)-20)/8
900 IF X<0 OR X>27 OR Y<0 OR
Y>23 THEN RETURN
910 IF SCR$(X,Y)<>"•" THEN R
ETURN
920 S=ASC(SCR$(X,Y,1)):IF S=
P+1 THEN RETURN
930 ERA I:LOCATE X,Y:PRINT
";:RETURN
940
950 FOR I=2 TO MA:IF MOVE(I)
OR F(I)=2 THEN 980
960 IF F(I)=1 THEN GOSUB 990
:GOTO 980
970 DEF MOVE(I)=SPRITE (C,RN
D(9),RND(3)+C/2,80,0):MOVE I
980 NEXT:RETURN
990
1000 F(I)=0:ERA I:POSITION I
,120,120:IF U<MA THEN F(I)=2
1010 U=U-1:IF U=0 THEN F(2)=
0:F(8)=1:RETURN
1020 IF U<0 THEN U=N:F(2)=0:
F(3)=0:F(4)=0:F(5)=0:F(8)=0:
C=C-2:L=L-1:IF C<2 THEN C=7:
IF M>-1 THEN S(1-M)=S(1-M)+1
1030 RETURN
1040 IF F(2) THEN I=2:GOTO99

```

●宇宙船が敵の基地などに
ぶつかった判定

●レーザーが敵の基地などに
あたった判定

●スピナーなどを出して動かす

●スピナーなどを命中させた。
あと後の処理

```

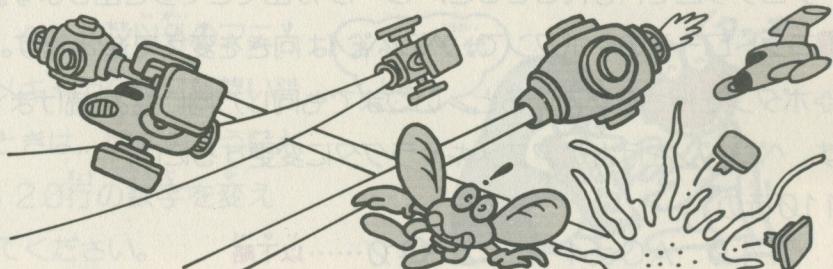
0
1050 IF MOVE(2)=0 THEN DEF M
OVE(2)=SPRITE(2,RND(8)+1,RND
(3)+1,10,0):MOVE 2
1060 RETURN
1070
1080 VIEW:RESTORE 1070
1090 FOR P=0 TO 200:LOCATERND(2
7),RND(23):PRINTCHR$(205):NE
XT
1100 LOCATE8,3:PRINT"STAR KI
LLER"
1110 FOR P=0 TO 1:LOCATE 10,
10+P*8:DEF MOVE(P)=SPRITE(P+
8,D(P),2,30):POSITION P,X0+P
*X1,Y0+P*Y1:MOVE P
1120 IF MOVE(P) THEN 1120
1130 READ A$:FOR I=1 TO LEN(
A$):PRINT MID$(A$,I,1);:PLAY
"C":NEXT:NEXT
1140 FOR P=2 TO 3:LOCATE 10,
17+P:READ A$:FOR I=1 TO LEN(
A$):PRINT MID$(A$,I,1);:PLAY
"C":NEXT:NEXT:LOCATE1,22:PRI
NT"フタリ テ" アソビ"マスカ(Y/N)":;M=-
1
1150 A$=INKEY$(0):IFA$="N" TH
EN M=1
1160 PAUSE80:RETURN
1170 DATA "PLAYER 1","PLAYER
2","OR","COMPUTER"

```

●ファイターフライを出して
うご動かす

●1人用2人用を選ぶ

サブルーチン



DEF MOVEとDEF SPRITE

ファミリーベーシックでは、アニメキャラクタを画面で動かすために、2つあります。1つはファミリーベーシックの大好きな特徴であるDEF MOVE命令を使う方法です。この方法だと、短いプログラムでアニメキャラクタに自然な動きをさせることができます。もう1つはDEF SPRITE命令を使う方法で、これは好みのキャラクタに自由な動きをさせることができます。

ここでは、その2つの方法でアニメキャラクタを動かすプログラムを紹介しましょう。2つのプログラムを見比べると、DEF MOVEとDEF SPRITEのそれぞれの特徴がわかつていただけると思います。



1. DEF MOVE (リスト2)

プログラムをRUNさせると、ペンペンが出てきて歩き出します。
①コントローラの+ボタンで、ペンペンは向きを変えて進みます。
+ボタンを押さないでいるときまでも同じ方向に進み続けます。ペンペンをほかのアニメキャラクタに変更するには、
110行の

110 AC=4 : AS=0……以下略

数字を0～15にすると、それに応じたアニメキャラクタになります。

```

10 'LIST 2
100 CLEAR : CLS : SPRITE ON
: CGSET 1,1
110 AC=4 : AS=0 : D=5 : X=11
6 : Y=116
120 POSITION 0,X,Y
200 =====
210 S=STICK(0) : IF S=0 GOTO
360
220 IF S=8 THEN D=1
230 IF S=9 THEN D=2
240 IF S=1 THEN D=3
250 IF S=5 THEN D=4
260 IF S=4 THEN D=5
270 IF S=6 THEN D=6
280 IF S=2 THEN D=7
290 IF S=10 THEN D=8
300 IF AS=D GOTO 360
310 AS=D
320 X=XPOS(0) : Y=YPOS(0)
330 DEFMOVE(0)=SPRITE(AC,D,1
,255,0,0)
340 POSITION 0,X,Y
350 MOVE 0
360 =====
370 IF MOVE(0)<>0 GOTO 410
380 DEFMOVE(0)=SPRITE(AC,D,1
,255,0,0)
390 POSITION 0,X,Y
400 MOVE 0
410 GOTO 200

```

リスト2

2. DEF SPRITE (リスト3)

ペンペン以外のアニメキャラクタにしたいときは、図9のように、120行の数字を変えください。

*RUNした後、①コントローラの+ボタンを押すと、アニメキャラクタが動きります。



120 P=96 : P1=P : P2=100
↑ 0 マリオ ↑
28 レディ 36
56 ファイターフライ 60
64 アキレス 68
88 ニタニタ 92
112 ファイアーボール 116
144 スピナー 148
184 カメさん 188
192 カニさん 196
200 トリさん 204

図 9

```

10 'LIST 3
100 CLEAR : CLS : CGSET 1,1
: SPRITE ON
110 X=116 : Y=116 : F=1
120 P=96 : P1=P : P2=100
200 '=====
210 K=STICK(0) : IF K=0 GOTO
380
220 AX=0 : AY=0
230 IF K=1 OR K=9 OR K=5 TH
EN AX=4 : F=1
240 IF K=2 OR K=6 OR K=10 T
HEN AX=-4 : F=-1
250 IF K=4 OR K=5 OR K=6 T
HEN AY=4
260 IF K=8 OR K=9 OR K=10 T
HEN AY=-4
270 IF F=1 THEN DEF SPRITE 0
,(0,1,0,1,0)=CHR$(P+1)+CHR$(P)
+CHR$(P+3)+CHR$(P+2)
280 IF F=-1 THEN DEF SPRITE
0,(0,1,0,0,0)=CHR$(P)+CHR$(P
+1)+CHR$(P+2)+CHR$(P+3)
290 X=X+AX
300 IF X>240 THEN X=0
310 IF X<0 THEN X=240
320 Y=Y+AY
330 IF Y>240 THEN Y=0
340 IF Y<0 THEN Y=240
350 SPRITE 0,X,Y
360 P=P+4 : IF P>P2 THEN P=P
1
370 PAUSE 3
380 '-----
390 GOTO 200

```

リスト 3

ファミリーベーシック

4. FAMILY

BASIC™ V3 の

めい れい はや

命令早わかり

角一始める行番号 (0~65534)

ループつづけのときも、財政を命令……

いくか (0~65534)

うつすり走るは西野、あらん様の命令……

下あり野食力由身地コラムのための

おもひき由自セーゼン、自貢の



ほんぶんよかた 本文の読み方

ここでは、ファミリーベーシックV3で新たに追加された BASIC の命令語について、解説します。解説文の読み方は、以下のようになります。

働き ……命令語の働きをひとことで説明したものです。

省略形 ……命令語を短縮したものです。

文法 ……命令語の使い方を一般的な形で表したものです。

- ・()の中は、ユーザーが自由に省略できます。
- ・{ }の中の項目は、ユーザーが自由に選択できますが、必ずどれかを選ばなくてはなりません。
- ・(), ' (カンマ)', ' (コロン)', ';' (セミコロン)', '- (ハイフン)', '= (イコール)'などの記号は、文法どおりに入力します。

説明 ……命令の使い方や注意点などが詳しく説明してあります。

……命令を使った例題や実行結果などが示されています。

……命令語の項目を省略したとき、自動的にセットされる値のことです。

[注意] TABとSPCは、拡張用予約語のため、変数名には使えません。

オート AUTO

働き 入力するプログラムの行番号を自動的に付けてくれます。

省略形 A.
文法 オート AUTO [m], [n]
m…始めの行番号 (0~65534)
n…いくつずつ行番号を増加して

いくか (0~65534)

デフォルト値 m=10 n=10

説明 これから入力するプログラムの始めの行番号と、いくつずつ増加させて行番号を付けるのか、その値を指定すると、自動的に行番号が付けられます。したがって、入力するのは、自動的にいた行番号の後のプログラム部分だけになります。1行入力が終わって **RETURN** キーを押すと、指定した増加分で次の行番号が付けられ表示されています。

m, n の指定を省略した場合は、デフォルト値として10が選択されます。プログラム中にすでにある行番号が表示された場合、前のプログラムを残すときは、

```

サンプル プログラム
** AUTO **
AUTO
OK
10 REM *AUTO
20 X=25
30 PRINT X
40 ←ここでストップキーをおす
OK
AUTO 100
OK
100 REM *AUTO
110 X=25
120 PRINT X
130 ←ここでストップキーをおす
OK
AUTO 100,30
100 REM *AUTO
130 X=25
160 PRINT X
190 ←ここでストップキーをおす
OK
AUTO , 50
190 REM *AUTO
240 X=25
290 PRINT X
340 ←ここでストップキーをおす
OK

```

リターン RETURN キーを押してその行を飛ばし（スキップ）ます。文字を入力してから RETURN キーを押すと、その行は新しい内容のものと入れ替わります。 STOP キーを押すと、 AUTO 命令は終了します。

バックアップ BACKUP

働き プログラムやBG面をバックアップするときに使います。

省略形 BA.

文法 BACKUP

説明 本体の電源をOFFにしても、ユーザーRAM領域の内容が消えてしまわないように、バッテリー（電池）により内容を残しておくための命令です。

プログラムと一緒にBG面1をバックアップしたい場合は、BG GET 命令を使ってあらかじめ、ユーザーRAM領域に転送しておく必要があります。

ただし、この場合はメモリ内のプロ

グラムが3Kバイト以下でなくてはなりません（BG GET の項目参照）。命令実行後、バックアップスイッチをONにするようメッセージが表示されます。

ページゲット BGGET

働き BG面1をバックアップするための命令です。

省略形 BGG.

文法 BG GET

説明 V RAM にあるBG面1のデータが、ユーザーRAMに転送されます。ただし、ユーザーRAMにメモリの余裕がない場合は、OMエラーが発生します（メモリ容量は4Kバイト）。

ページセット BGPUS

働き BG GET によりバックアップされたBG面を、V RAM へ転送する命令です。

省略形 BGP.
文法 ビジー ブット
説明 BG GET によりユーザー RAM に
転送されたデータを, V RAM に転
送します。すでに転送済みの場合は,
NB エラーが発生します。

ホット スタート ピー ジー ブット じ
また, HOT START の場合は, 自
動的に転送され, BG PUT 実行後
と同じ状態になります。

BGTOOL

働き 背景画を描くための, モードに移行
する命令です。
省略形 BG.
文法 ビジー ツール
説明 BG面1で背景画を描くために,
BASIC モードからBGグラフィックモードへ移行します。これは, プログラムを作成している途中でも実行することができます。また, この命令をプログラムに組み込むこともできます。BGグラフィックの作成方法は,これまでと同じです。なあ,

ベーシック BASIC モードに戻るには, エスケープ キーを押した後, STOP キーを押します。

CAN

働き DEF MOVE で動きを定義したア
ニメキャラクタを, 未定義にします。

文法 CAN n0, n1, n2, ..., n7
n0~n7…スプライト番号

説明 DEF MOVE によって定義したア
ニメキャラクタの, 表示と座標位置
の両方を同時に未定義にする命令です。

これまでのファミリーベーシックでは, アニメキャラクタの表示を消すとともに, 座標位置を未定義(0,0)にするには, ERA 命令と POSITION 命令の2つが必要でした。V3 では, この CAN 命令だけで, これ

を実行することができます。

クリック CLICK

**はたら
働き** キー入力音(クリック音)のON/OFFを行ないます。

省略形 CLI.O./CLI.OF.

文法 CLICK {ON}...クリック音発生
CLICK {OFF}...無音

シー エル エス CLS

**はたら
働き** BG面の表示をクリアします。

省略形 CL.

文法 CLS(BG面)
0...BG面のクリア
1...BG面1のクリア
ただし、省略の場合は現在アクセス中のBG面がクリアされる。

説明 BG面には、0と1の2面があります。どちらの画面をクリアするかを番号で指定します。

できます。BGグラフィックの作成方法は、これまでと同じです。なあ、

クラッシュ CRASH

**はたら
働き** DEF MOVEによるアニメキャラクタ同志の重なりを判定します。

省略形 CR.

文法 CRASH(n)...スプライト番号(0~7)

説明 アニメキャラクタ同志の重なりを判定します。

アニメキャラクタが重なっている場合、CRASH(n)は、重なっているスプライトのうちで、いちばん小さなスプライト番号となります。

ただし、重なっていない場合は、マイナス(-1)となり、またスプライトが未定義の場合は、-2となります。

```

1 1 サンプル フログラム
5 SPRITE
10 VIEW
20 SPRITE ON
30 CGSET 1,0
40 D=3
50 FOR N=0 TO 7 :DEF MOVE(N)
=N=SPRITE(N,D,N+1,255,0):NEXT
60 MOVE 0,1,2,3,4,5,6,7
70 LOCATE 0,3 :PRINT "VCT(0)";
;VCT(0)
80 LOCATE 0,5 :PRINT "XPOS(0)",
,YPOS(0); XPOS(0); ; ; ;Y
POS(0); ;
90 LOCATE 0,7 :PRINT "CRASH(0)";
; ; ;CRASH(0)
100 LOCATE 0,1 :PRINT "MOVE(0)";
; ;MOVE(0)
110 IF MOVE(0)=-1 THEN 70
120 D=RND(8)+1 :PLAY "T104B1A
G2FE"
130 GOTO 50

```

テリート DELETE

**はたら
働き** プログラムの中で消したい部分があるとき、その行番号を指定すること

によって、まとめて取り消すことが
できます。

省略形

DEL.

文法

DELETE [m] { , } [n]

m … 取り消しを始める行番号

n … 取り消しを終える行番号

説明

mの値だけを指定した場合は、その

行だけが取り消されます(②)。また、

- (ハイフン)から n の値だけを指定

した場合は、プログラムの先頭から

指定行(n)までが、取り消されます

(③)。nの値を省略した場合は、m

の値から最後の行までが、取り消さ

れます(④)。

DELETE m - n … ①

DELETE m … ②

DELETE - n … ③

DELETE m - … ④

```

サンプル フロッキラム
** DELETE **
LIST
10 REM *DELETE
20 INPUT "A,B,X" ;A,B,X
30 INPUT "A,B,X" ;A,B,X
40 Y=A*X+B
50 Y=A*X+B
60 Y=A*X+B
70 PRINT "A*X+B=" ;Y
80 PRINT "A*X+B=" ;Y
90 PRINT "A*X+B=" ;Y
100 END
OK
DELETE 30,50
OK
LIST
10 REM *DELETE
20 INPUT "A,B,X" ;A,B,X
60 Y=A*X+B
70 PRINT "A*X+B=" ;Y
80 PRINT "A*X+B=" ;Y
90 PRINT "A*X+B=" ;Y
100 END
OK
DELETE 80,
OK
LIST
10 REM *DELETE
20 INPUT "A,B,X" ;A,B,X
60 Y=A*X+B
70 PRINT "A*X+B=" ;Y
OK

```

イーアールアール
ERL

働き エラーの発生した行番号が求められ
ます。

説明 エラーが発生したとき、どの行で発

生したのか、その行番号を知るため

に使われます。

エラーが発生すると, ON ERROR

GOTO 文によってエラー処理ルー

チンに移り, ERL (エラーの起きた

行番号) の値によって処理を決定す

るというように用いることができます。

イーアールアール
ERR**働き** エラーが発生したとき、そのエラー

コードが求められます。

文法 イーアールアール
ERR**説明** エラーが発生すると, ON ERROR

GOTO 文によってエラー処理ルー

チンに移り、その中でエラーの処理

が行なわれます。

イーアールアール
ERRは、エラーの内容に対応した

エラーコードがその値となり、エラ

ーの種類により、処理内容を決定す

るために使われます。

*ON ERROR GOTO の項のサンプルプロ
グラムおよびエラーコード表(P.127)参照

ERROR

働き 仮のエラーを発生させます。

省略形 ERR.

文法 ERROR エラーコード

*エラーコード表(P.127)参照

説明 これは、エラー処理ルーチンが正しく働くかどうかを確認するために、仮のエラーを発生させる命令です。指定したエラーコードに対応するエラーが、その行番号で発生します。

FIND

働き プログラムの中から、指定した文字列を探し、それを含む行を表示させる命令です。

省略形

文法 FIND "X"

X…探ししたい単語 31字以内

説明

これは、1度入力した行に対して行なうことのできる命令です。プログラムを変更するときなどにこの命今を使うと、変更したい単語(文字列といいます)を含む行をすべて表示してくれるので、とても便利です。

FILTER

働き BG面の全面着色が実行されます。

省略形 FIL.

文法 FILTER カラー番号

0番	1番	2番	3番	4番	5番	6番	7番
無色	赤	緑	黄	青	マゼンタ	空色	白

ただし、デフォルト値は0
フィルターをかけたように、淡く着色されます。

```
サンプル プログラム
** FIND **
LIST
10 REM *SEARCH
20 INPUT A,B,X
30 Y=A*X+B
40 PRINT A;B;X;Y
50 END
OK
FIND "A"
10 REM *SEARCH
20 INPUT A,B,X
30 Y=A*X+B
40 PRINT A;B;X;Y
50 END
OK
FIND "A"
20 INPUT A,B,X
40 PRINT A;B;X;Y
OK
```

色の指定番号は、表のとおり0~7の8種類です。何も番号を指定しない場合は、デフォルト値(自動的にセツトされる値)として、0が選択されます。

また、**CTR**と**D**キーを同時に押してもデフォルト状態になります。

ゲーム GAME

ROMに内蔵されているゲームが実行されます。

省略形 GA.

文法 GAME P [, Sw]

P…ゲーム番号 0~3
Sw…背景画の転送を行なう：1
転送しない：0または省略

説明 ゲームは4種類あります。呼び出すときは、それぞれのゲーム番号で指定します。
この命令は、プログラム中に組み込むことはできません。

なあ、GAMEを呼び出すと、その前に記憶されていたプログラムは、自動的に消えてしまいますから、注意してください。

V3カセットに内蔵されているゲームのうち、背景画の転送の必要があるのは、『マリオワールド』だけです。

インストリング INSTR

ROM中の文字列1の中で、指定した文字(文字列2)の位置が求められます。

省略形 INS.

文法 INSTR ((P,) 文字列1, 文字列2)

P…文字列1を探し始める位置。

説明 デフォルト値は1。
まず、文字列1の中に文字列2が含まれているかどうかを探します。含まれていればその位置を、文字列1の先頭から数え、その数字が値となる。

```

サンプル プログラム
5 REM *INSTR
10 A$="NINTENDO HUBASIC"
20 PRINT INSTR(A$, "TEN")
RUN
4

```

ります。値の範囲は1~31で、文字列がみつからなかつた場合の値は0となります。また、位置の数え方は左から右です。

されます。一でるハルメル内コイドサビと
また、圖面と書体要領の表記の画意圖、さこの△

ロード エス LOADS

働き カセットテープからBGグラフィックのデータを呼び出す命令です。

文法 LOADS "ファイル名"

ただし、ファイル名：31文字以内

説明 プログラムをロードする命令 (LOAD AD) と、基本的には同じです。

ファイル名を指定すると、その名前のBGが見つかるまで、他のBGやプログラムを読み飛ばし (SKIPと表示され) ます。

ファイル名を省略すると、カセットテープの1番目に入っているBGが呼び出されます。

なお、プログラムとBGと一緒に呼び出したい場合は、

ロード LOADS "プログラム名":LOAD

エス ファイル名 RETURN

とします。

ロード中は、画面にLOADINGファイル名と表示され、呼び出しが終了するとOKとなります。

オン エラー ゴーツー ON ERROR GOTO

働き エラーが発生したとき、分岐する行を指定します。

省略形 O. ERR. G.

文法 ON ERROR GOTO 行番号

説明 プログラムの先頭部分で、このステートメントを実行することにより、以降の行でエラーが発生した場合、指定行 (エラー処理ルーチン) へ分岐します。

分岐先ルーチンの終了行は、

RESUME または END 命令に

しておきます。RESUME の場

合は、メインルーチンのあらか

じめ指定しておいた行に再び戻

ります。また、END の場合は、

```
1 'サンプル プログラム
5 /*ON ERROR GOTO/RESUME/
6 REM *ON ERROR SAMPLE
10 ON ERROR GOTO 100
15 INPUT "A=" ; A
20 INPUT "B=" ; B
25 C=A/B
30 PRINT "A/B=" ; C
35 GOTO 30
40 PRINT "トケマセン"
45 PRINT "ERR=" ; ERR
50 PRINT "ERL=" ; ERL
55 RESUME 30
```

プログラムは終了し、コマンド
の入力待ちとなります。

メインルーチン
ON ERROR
GOTO

メインルーチン

エラー発生

(分岐)

エラー処理
ルーチン

メインルーチン
へ戻る

終了

RESUME
END

リナンバー RENUM

働き プログラムの行番号を付け替える命
令です。

REN.

省略形

RENUM [L], [m], [n]

L…新しくつける開始行番号

m…変更を始める行の元の行番号
n…いくつごとに行番号を増加し
ていくのか。

デフォルト値 L, m, n 共に10

説明 プログラムの行番号を整理する命令

です。行番号が整理されるだけでな
<, GOTO, GOSUB 文などの分
岐先行番号も、新しい行番号に対応
して自動的に変更されます。

ただし、ERL=1000のように、定数
で指定した行番号は変更されません
ので注意してください。

サンプル プログラム
*** RENUM ***
5 REM *RENUM
20 FOR I=0 TO 10
21 PRINT "BASIC V3."
55 NEXT
RENUM

OK

LIST

10 REM *RENUM
20 FOR I=0 TO 10
30 PRINT "BASIC V3."
40 NEXT

OK

RENUM 100

OK

LIST
100 REM *RENUM
110 FOR I=0 TO 10
120 PRINT "BASIC V3."
130 NEXT

OK

RENUM 20 , , 20

OK

LIST

20 REM *RENUM
40 FOR I=0 TO 10
60 PRINT "BASIC V3."
NEXT
OK

RESUME

働き エラー分岐ルーチンからメインル
ーチンに戻す命令です。

RESU.

省略形

RESUME [(行番号)]
[(次回)NEXT]

RESUMEのみ …エラー発生行
から再び実行
…指定した行番

行番号

号から実行
NEXT

…エラー発生行
の次の行から

実行

説明 ON ERROR GOTO 文によって
分岐したエラー処理ルーチンから,
メインルーチンに戻す命令です。

*ON ERROR GOTO の項のサンプルプログラム参照

セーブ エス SAVES

働き カセットテープにBGグラフィック
のデータを保存する命令です。

文法 SAVES "ファイル名"

ただし, ファイル名 31文字以内

説明 プログラムをセーブする命令 (SA
VE) と基本的には同じです。
プログラムとBGと一緒に保存して
おくには,

セーブ
SAVE "プログラム名":SA

エス
VES "ファイル名" RETURN

とします。

セーブ中は, 画面に WRITING ファ
イル名と表示され, 読み込みが終了
すると OK となります。

SCREEN

働き 表示面とアクティブ面に, 2面ある
BG面のいずれかを割り当てます。

省略形 SC.

文法 SCREEN 表示面(, アクティブ面)

命令	表示面	アクティブ面
SCREEN 0, 0	0	0
SCREEN 0, 1	0	1
SCREEN 1, 0	1	0
SCREEN 1, 1	1	1
[CTR]+D	0	0

*この数字はBG面のと1を表しています。

説明 BG面を2面持っていますので、表示面とアクティブ（書き込み）面をそれぞれ別々に指定できます。表示面とアクティブ面が同一の場合、アクティブ面の指定を省略することができます。

スクリーン SCREEN Ø, Ø → SCREEN Ø
コントロールキーを押しながら□キーを押すと、表示面とアクティブ面は共にBG面Øになります。

※アクティブ面とは、カーソルが表示されており、カーソルを動かして、実際に表示を変更できる面のことをいいます。

トレスオン TRON / TROFF

働き トレスモードを実行・解除します。
文法 TRON…トレス実行モード
TROFF…ノーマルモードに戻る
説明 TRON文により、プログラム実行中の行番号を表示させることができます。

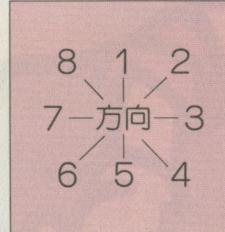
トレスオン ぶんじっこう TRON文の実行はダイレクトモードでもプログラムモードでもできます。TROFFにすると、TRON命令は終了し、元のモードに戻ります。

```
サンプル フロクラム
5 REM *TRON/TROFF
10 FOR N=0 TO 9
20 PRINT "TR=";N
30 NEXT
**じっこうけっかはつぎのとおりです**
#10 #20 TR=0
#30 #20 TR=1
#30 #20 TR=2
: : : :
: : : :
#30 #20 TR=9
#30
```

ベクター VCT

働き DEF MOVEによるアニメキャラクタの移動方向を読み取ります。
省略形 VC.
文法 VCT(n)

n…スプライト番号 0~7
VCT(n) の値 0~8



説明 アニメキャラクタが動作しているときに、その方向を番号で与えます。

ベクター
VCT(n)の値は、DEF SPRITE文

における方向パラメータと同じ値で口でます。

す。ただし、アニメキャラクタが停止します。

止している場合や、未定義での値は

0です。タイプ面の指定を出題する

* CRASH の項のサンプルプログラム参照

SCREEN 0,0 → SCREEN 0

キーを押しながら圆キーを

出題する

ことをいいます。

アクティベーションモード

示されており、あるいは娘や姫に向式便券の上で

て、実際に表示を変更できる面の

ことをいいます。

VCT(n)文

→ Q 値番号でス…n

8~0 間の(n)T0V

TRON/T0V

トレー モードで実行します。

トロント

TROFFノーマルモードに戻る

トロントにより、プログラム実行

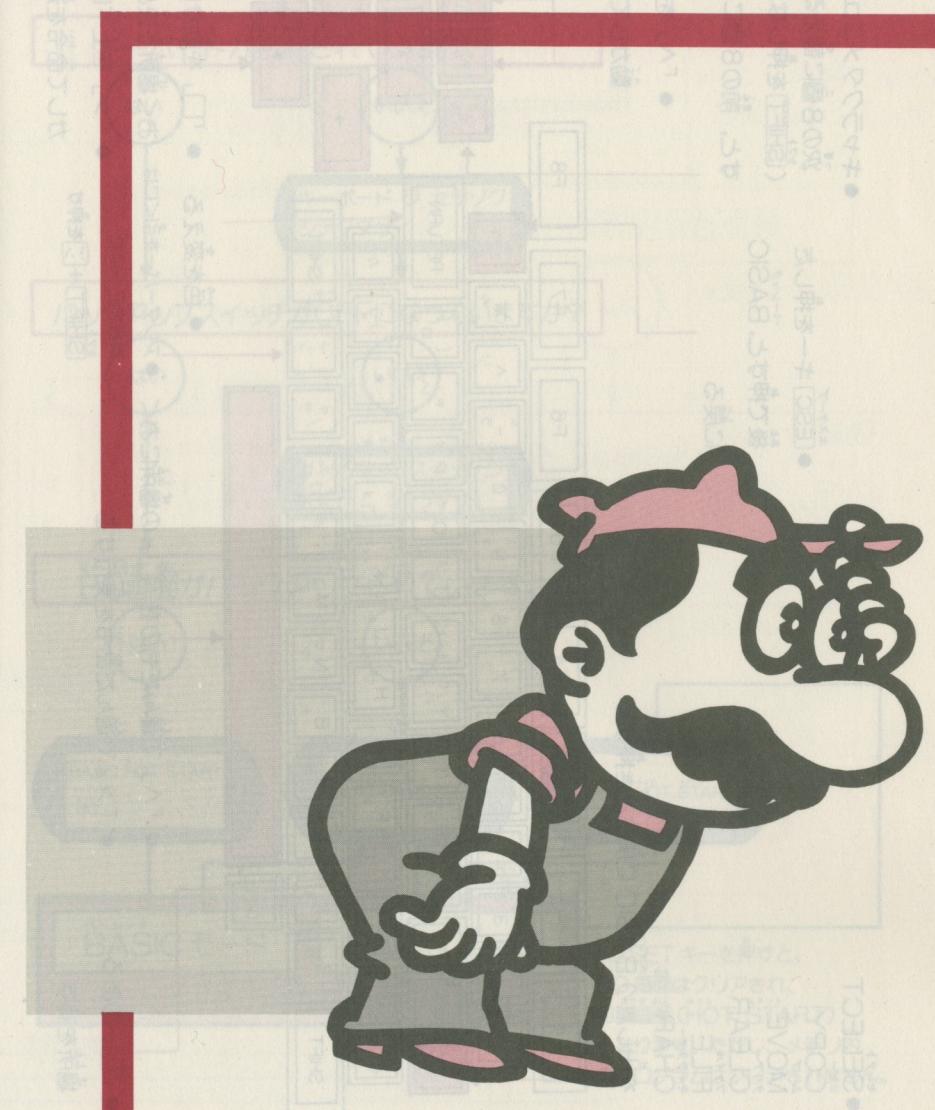
中の行番号を読み上げて表示します。

もしも大老ケ是番に向式のチコキ

ファミリーベーシック

FAMILY BASIC™ V3

ふろく

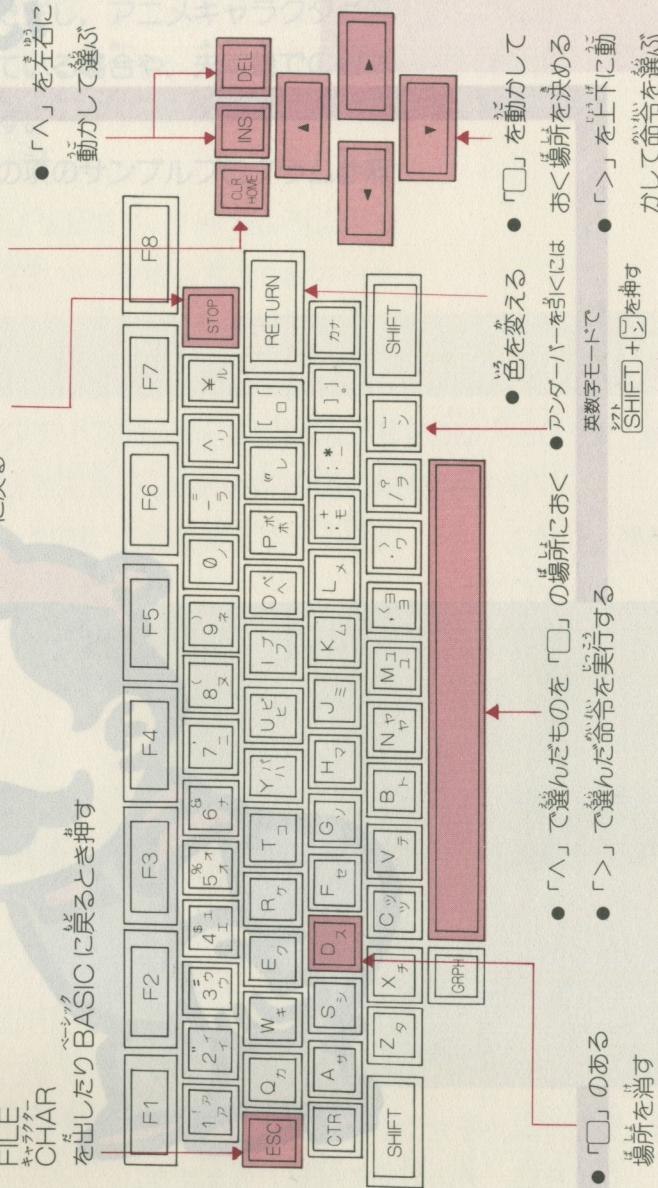


BG TOOLでのキーの動き

- キヤラクタメニューを
次の8箇目に替える
ソフト([SHIFT])を押ししながら押すと、前の8箇目になる

● エスケープ [ESC] キーを押した
あとで押すと、BASIC が止まる

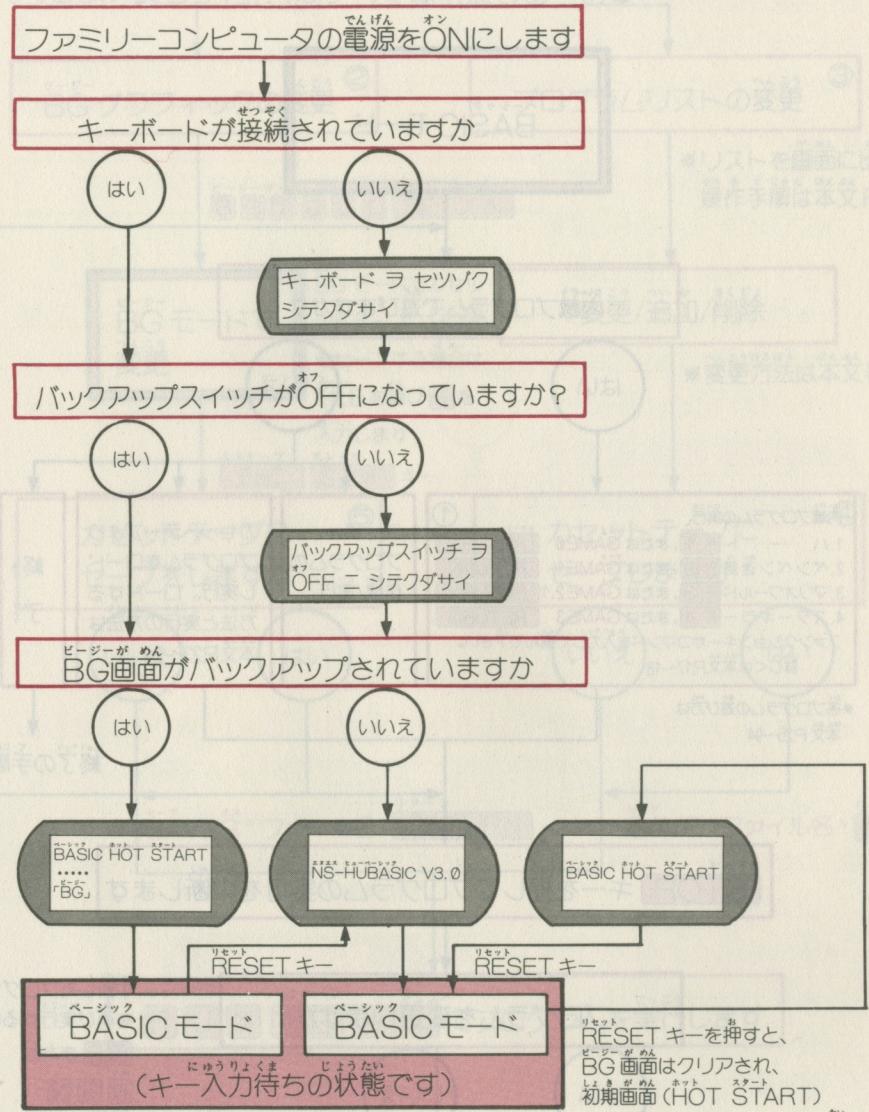
セレクト SELECT ●
コピ COPY
ムーブ MOVE
クリア CLEAR
ファイル FILE
キャラクタ CHAR
ベイビーベイビーベイビー^{を出したり}
BASICに戻るとき押す



120

スタート(電源ONから)の手順

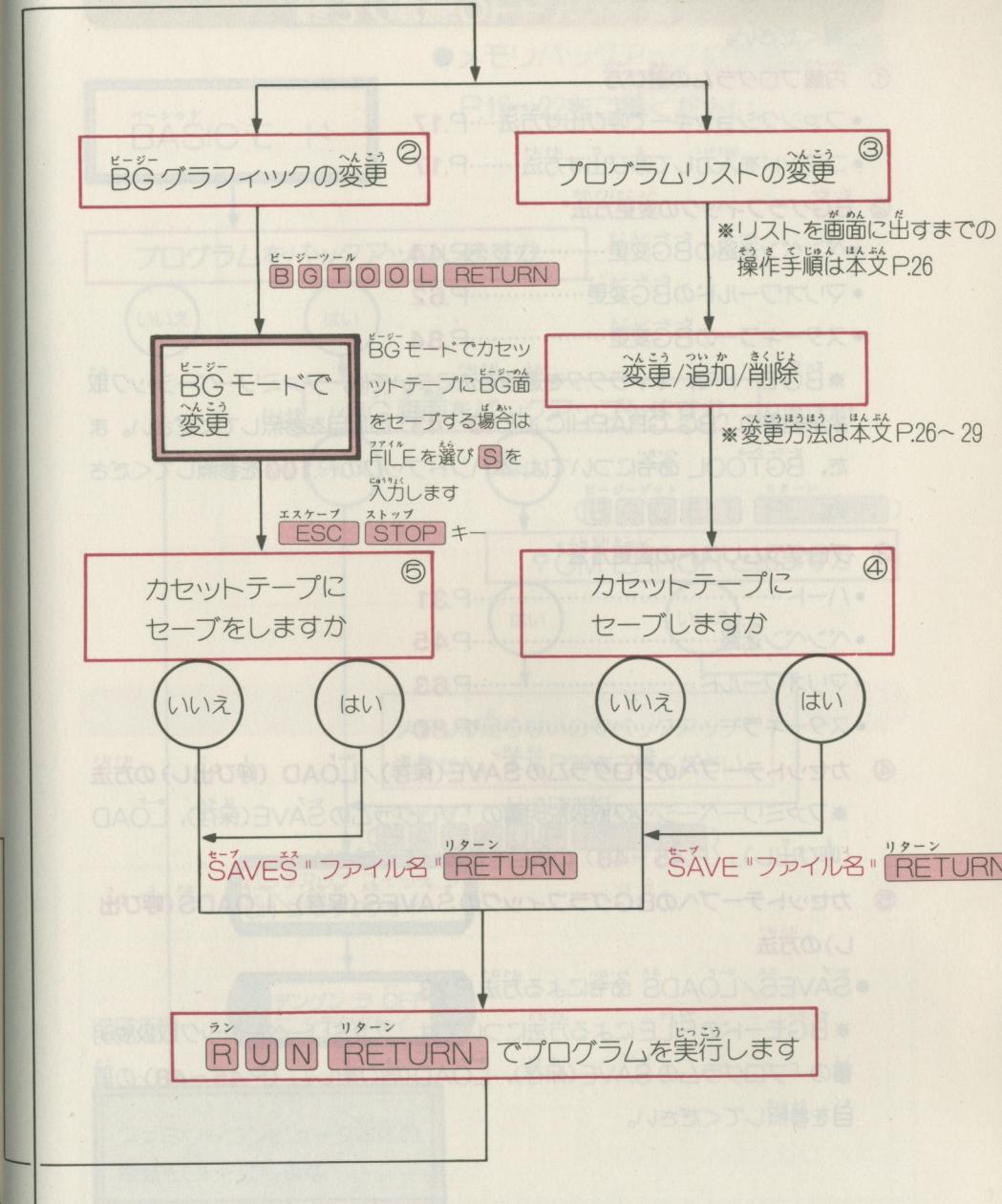
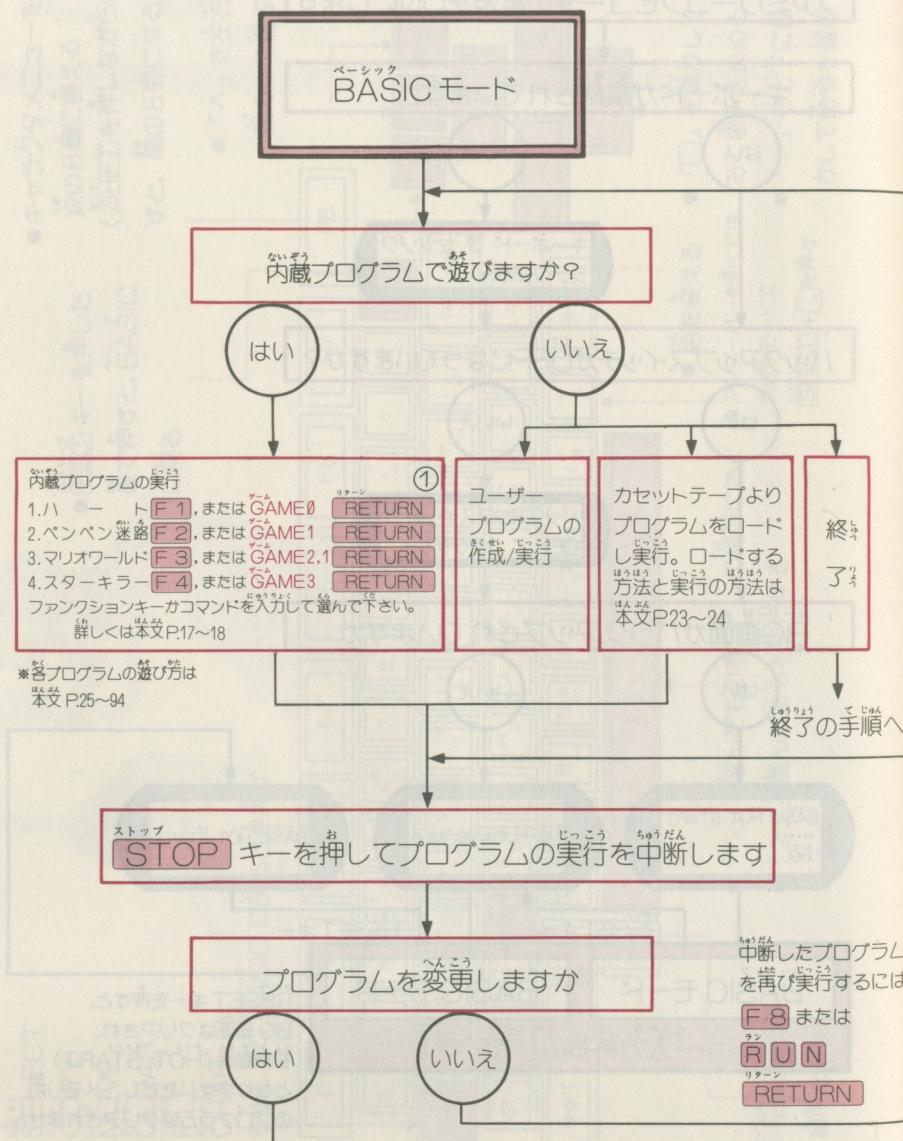
●本文P.14~16をご覧ください。



リセット
RESET キーを押すと、
ビージーがめん
BG 画面はクリアされ、
しょきがめんホットスタート
初期画面(HOT START)
となります。ただし、メモリ内
のプログラムはクリアされません。

内蔵プログラムの実行と変更

●①～⑤の操作についてはP.124をご覧ください。



- 前ページの図のうち、番号のついた操作については、以下のページを、それぞれご覧ください。

① 内蔵プログラムの選び方

- ファンクションキーで呼び出す方法…P.17
- コマンドを入力して呼び出す方法…P.17

② BGグラフィックの変更方法*

- ペンペン迷路のBG変更…P.44
- マリオワールドのBG変更…P.62
- スターキラーのBG変更…P.84

*BGモードで、キャラクタを選ぶ方法については、ファミリーベーシック取扱説明書の「BG GRAPHIC」(P.40～43)の項目を参照してください。また、BGTOOL 命令については、本ハンドブックのP.100を参照してください。

③ プログラムリストの変更方法

- ハート…P.31
- ペンペン迷路…P.45
- マリオワールド…P.63
- スターキラー…P.80

④ カセットテープへのプログラムのSAVE(保存)/LOAD(呼び出し)の方法

*ファミリーベーシック取扱説明書の「プログラムのSAVE(保存), LOAD(呼び出し)」(P.45～48)の項目を参照してください。

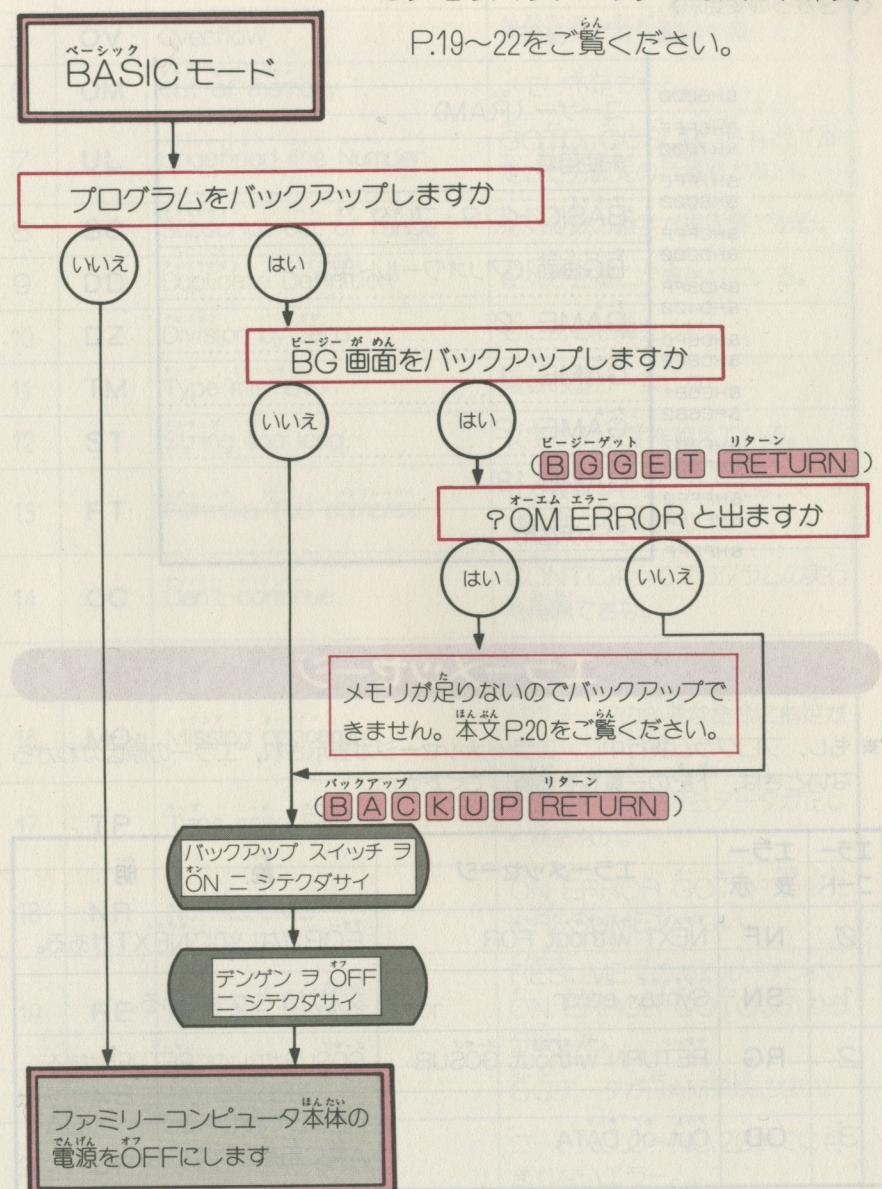
⑤ カセットテープへのBGグラフィックのSAVES(保存)/LOADS(呼び出し)の方法

• SAVES/LOADS命令による方法 P.23

*BGモードのFILEによる方法については、ファミリーベーシック取扱説明書の「プログラムのSAVE(保存), LOAD(呼び出し)」(P.45～48)の項目を参照してください。

終了(電源OFF)の手順

- メモリ/バックアップの方法は、本文P.19～22をご覧ください。



メモリマップ

〈V3からの変更点〉

&H6000	ユーザー (RAM)
&H6FFF	未使用域
&H7000	
&H7FFF	
&H8000	BASIC インタ・プリタ
&HCFFF	
&HD000	BG画面 (マリオワールド用)
&HD3FF	
&HD400	GAME Ø
&HDBFD	
&HDBFE	GAME 1
&HE681	
&HE682	GAME 2
&HF307	
&HF308	GAME 3
&HFFFF9	
&HFFFA	システム用
&HFFFF	

エラーメッセージ

*もし、プログラム実行中に、エラーメッセージが表示され、エラーの原因がわからないときは、下記の一覧表を参照してください。

エラー コード	エラー 表示	エラーメッセージ	説 明
Ø	NF	ネクスト ウィズアウト フォー NEXT without FOR	フォー FORがないのにNEXTがある。
1	SN	シンタックス エラー Syntax error	文法がまちがっている。
2	RG	リターン ウィズアウト ゴーサブ RETURN without GOSUB	ゴーサブ GOSUBがないのにRETURNがある。
3	OD	アウト オブ データ Out of DATA	リード READで読むべきデータがDATA文に用意されていない。

4	IL	イリーガル ファンクション コール Illegal function call	ステートメントや関数の呼び方がまちがっている。
5	OV	オーバーフロー Overflow	演算結果が許容範囲を越えた。
6	OM	アウト オブ メモリー Out of memory	メモリ不足である。
7	UL	アンデファインド ライン ナンバー Undefined line Number	ゲーム GOTO, GOSUB, IFなどで指定した分岐先の行番号がない。
8	SO	サブスクリプト アウト オブ レンジ Subscript out of range	配列変数の添字が規定外である。
9	DD	デュプリケイト デフィニション Duplicate Definition	配列が2重に定義されている。
10	DZ	デビジョン バイ ゼロ Division by zero	0による除算をした。
11	TM	タイプ ミスマッチ Type mismatch	変数の型が一致しない。
12	ST	ストリング ツー ロング String too long	文字が31文字を越えている。
13	FT	フォーミュラ ツー コンプレックス Formula Too complex	式が複雑すぎる。たとえば、()が異常に多い場合。
14	CC	キャント コンティニュー Can't continue	CONTによってプログラムの実行を再開できない。
15	UF	アンデファインド ファンクション Undefind function	未定義の関数を呼んだ
16	MO	ミッシング オペランド Missing operand	パラメータの必要な命令に指定がない。
17	TP	テーブ リード エラー Tape read ERROR	カセットテープからデータが正しく読めない。
18	NR	ノーリジューム No RESUME	ON エラー GOTO で分岐先にRESUME文がない。
19	RE	リジューム ウィズアウト エラー RESUME without error	RESUME文のあるルーチンへON エラー GOTO以外から分岐した。
20	NB	ノーピージーデータ No BG data	BGデータがRAM領域にない。
21	UP	アンプリントブル エラー Unprintable error	コードNO.Ø~NO.2Øまでに含まれないエラー。

*コードNO.15, 18~21はV3で新たに追加されたものです。

スプライトの関数値とアニメキャラクタの状態の対応表

アニメキャラクタの状態					
関数	未定義	定義時	動作中	停止	
ムーブ MOVE(n)	Ø	Ø	-1	Ø	
クラッシュ CRASH(n)	-2	1又はØ	-1又はØ~7	-1又はØ~7	
ベクター VCT(n)	Ø	Ø~8	Ø~8	Ø~8	
エックスポス XPOS(n)	Ø	120	(x,y)		
ワイポス YPOS(n)	Ø	120	(x,y)		

アニメキャラクタや、BGグラフィックで使用するキャラクタの色
 形については、これまでのファミリーベーシック「取扱説明書」
 P.113、裏表紙およびキャラクタマップをご覧ください。

*16種類のアニメキャラクタには、次の数字が割り当てられています。
 改造をするときこの数字を入れ替えると、アニメキャラクタが
 替わります。

Ø : マリオ	6 : 車	12 : レーザー
1 : レディ	7 : スピナー	13 : シェルクリーパー（カメさん）
2 : ファイターフライ	8 : スターキラー	14 : サイドステッパー（カニさん）
3 : アキレス	9 : スターシップ	15 : ニットピッカー（トリさん）
4 : ペンペン	10 : 爆発	
5 : ファイアーボール	11 : ニタニタ	

DEF MOVE系とDEF SPRITE系の命令対照表

機能	分類
DEF SPRITE系 コーネザーが定義する アニメキャラクタ	DEF MOVE系 特定の動きをする アニメキャラクタ
表示を可能にする 表示を不可能にする	SPRITE ON SPRITE OFF
定義する	DEF SPRITE ~
未定義の状態にする	RUN 命令での初期化
動かす、または表示する	SPRITE n, X, Y
動きを止める	SPRITE n, X, Y
消す	SPRITE n
初めの位置を決める	SPRITE n, X, Y
現在の位置を知る	POSITION n, X, Y
動いているか止まっているか 動いているときの方向を知る	XPOS(n), YPOS(n)
重なりの判定	MOVE(n) VCT(n) CRASH(n)
ユーザーのプログラムにより実現できる	

ビージー BGキャラクタをリストに使う方法

内蔵プログラムのリストを見ると、リスト中に直接BGキャラクタが描き込まれている部分があります。

たとえば「ハート」(GAMEØ)の300行の途中にある、

PRINT "●"

などです。このようなBGキャラクタは、直接キーで打ち込むことができません。BGキャラクタをリストに使うときには、次のようにしてください。

たとえばハタのBGキャラクタを使った

250 PRINT "P"

という行を例にしましょう。まず、行番号を付けずに

PRINT CHR\$(199) RETURN

とキーを押します。これで画面にPが出てきます。そうしたらPの上にカーソルを動かして、INSキーを10回ほど押してください。これで押した回数だけ、Pが右に動いていきます。適当なところまで右に動いたら、

250 PRINT

とキーを押します。このときPが消えてしまう場合は、さらにINSキーを押して、Pを右に動かしてください。ここまで操作で、画面が

250 PRINT

となつていればいいのですが、たいていはPが少し右に行きすぎています。そのときはカーソルをPの上に動かして、ちょうど"'"の右にくるまでDELキーを押してください。

最後にPのすぐ右に、"'"を付ければでき上がりです。

最初に打ち込んだ

PRINT CHR\$(199)

の199を184～255の間でいろいろ変えると、ほかのBGキャラクタをリストに使うことができます。数字とBGキャラクタの関係は、付録のキャラクターテーブル(P.131)を参照してください。

キャラクターテーブルB

*キャラクターテーブルA、およびA～Dを含むキャラクターテーブル

Bは、ファミリーベーシック取扱説明書をご覧ください。

	Ø	1	2	3	4	5	6	7
E	—							
	184	185	186	187	188	189	190	191
F								
	192	193	194	195	196	197	199	199
G								
	200	201	202	203	204	205	206	207
H								
	208	209	210	211	212	213	214	215
I								
	216	217	218	219	220	221	222	223
J								
	224	225	226	227	228	229	230	231
K								
	232	233	234	235	236	237	238	239
L								
	240	241	242	243	244	245	246	247
M								
	248	249	250	251	252	253	254	255

キャラクターコード表

ファミリーベーシックのキャラクタコードを、一覧表で示します。BASICのCHR\$(n)、ASC("A\$")による文字とコードの変換は、これに対応しています。キャラクタコード表A、またはBのキャラクタ、文字、記号は、パップラウンド、スプライトの両面で使用できます。(どの面にどちらの表を使用するかの指定はCOSTUME(kostume)を使用します)

●キャラクタコード表A（主に、スプライト用）

コード(10進数)は、キャラクターテーブルAでアニメキャラクタの四隅に表記してある番号に対応します。取扱説明書・裏表紙参照】

コード (10進数)	コード (16進数)	説明	コード (10進数)	コード (16進数)	説明	コード (10進数)	コード (16進数)	説明	コード (10進数)	コード (16進数)	説明
0	00		32	20		64	40		96	60	
1	01	マリオ (WALK1)	33	21	レディ (WALK2)	65	41	アキレス (左1)	97	61	ベンベン (左歩1)
2	02		34	22		66	42		98	62	
3	03		35	23		67	43		99	63	
4	04		36	24		68	44		100	64	
5	05	マリオ (WALK2)	37	25	レディ (WALK3)	69	45	アキレス (左2)	101	65	ベンベン (左歩2)
6	06		38	26		70	46		102	66	
7	07		39	27		71	47		103	67	
8	08		40	28		72	48		104	68	
9	09	マリオ (WALK3)	41	29	レディ (JUMP)	73	49	アキレス (左上) (1)	105	69	ベンベン (正面)
10	0A		42	2A		74	4A		106	6A	
11	0B		43	2B		75	4B		107	6B	
12	0C		44	2C		76	4C		108	6C	
13	0D		45	2D	レディ (スリップ) (着地)	77	4D	アキレス (左上) (2)	109	6D	ベンベン (後)
14	0E	マリオ (JUMP)	46	2E		78	4E		110	6E	
15	0F		47	2F		79	4F		111	6F	
16	10		48	30		80	40		112	70	
17	11	マリオ (スリップ) (着地)	49	31	レディ (はしご)	81	51	アキレス (上1)	113	71	ファイアーポール(1)
18	12		50	32		82	52		114	72	
19	13		51	33		83	53		115	73	
20	14		52	34		84	54		116	74	
21	15	マリオ (はしご)	53	35	レディ (DOWN)	85	55	アキレス (上2)	117	75	ファイアーポール(2)
22	16		54	36		86	56		118	76	
23	17		55	37		87	57		119	77	
24	18		56	38		88	58		120	78	
25	19	マリオ (DOWN)	57	39	ファイター フライ(1)	89	59	ニタニタ (1)	121	79	車 (左1)
26	1A		58	3A		90	5A		122	7A	
27	1B		59	3B		91	5B		123	7B	
28	1C		60	3C		92	5C		124	7C	
29	1D	レディ (WALK1)	61	3D	ファイター フライ(2)	93	5D	ニタニタ (2)	125	7D	車 (左2)
30	1E		62	3E		94	5E		126	7E	
31	1F		63	3F		95	5F		127	7F	

- アニメキャラクタを定義する場合は、このコード表にあるコード(10進数、または16進数)で指定してください。
- 0～255までの10進と16進の換算表としてもご利用ください。

コード (10進数)	コード (16進数)	説明	コード (10進数)	コード (16進数)	説明	コード (10進数)	コード (16進数)	説明	コード (10進数)	コード (16進数)	説明
128	80	車 (左上1)	160	A0	スターキラー (上)	192	C0	サイド ステッパー (1)	224	E0	はねベン (1)
129	81		161	A1		193	C1		225	E1	
130	82		162	A2		194	C2		226	E2	
131	83		163	A3		195	C3		227	E3	
132	84	車 (左上2)	164	A4	スターシップ (左)	196	C4	サイド ステッパー (2)	228	E4	はねベン (2)
133	85		165	A5		197	C5		229	E5	
134	86		166	A6		198	C6		230	E6	
135	87		167	A7		199	C7		231	E7	
136	88	車 (上1)	168	A8	スターシップ (左上)	200	C8	ニット ピッカー (1)	232	E8	はねベン (1)
137	89		169	A9		201	C9		233	E9	
138	8A		170	AA		202	CA		234	EA	
139	8B		171	AB		203	CB		235	EB	
140	8C	車 (上2)	172	AC	スターシップ (上)	204	CC	ニット ピッcker (2)	236	EC	はねベン (2)
141	8D		173	AD		205	CD		237	ED	
142	8E		174	AE		206	CE		238	EE	
143	8F		175	AF		207	CF		239	EF	
144	90	スピナー (1)	176	B0	爆発 (1)	208	D0	レーザー (よこ)	240	F0	1
145	91		177	B1		209	D1	レーザー (ななめ)	241	F1	2
146	92		178	B2		210	D2		242	F2	3
147	93		179	B3		211	D3		243	F3	4
148	94	スピナー (2)	180	B4	爆発 (2)	212	D4	レーザー (たて)	244	F4	COM PUTER 文 の字
149	95		181	B5		213	D5	スプライト用 ぬりつぶし の色 (1),(2),(3)	245	F5	
150	96		182	B6		214	D6		246	F6	
151	97		183	B7		215	D7		247	F7	
152	98	スターキラー (左)	184	B8	シェル クリーパー (1)	216	D8	ミュージック ボード用 カーソル (1),(2),(3)	248	F8	OP ER 文 の字 A T O R
153	99		185	B9		217	D9		249	F9	
154	9A		186	BA		218	DA		250	FA	
155	9B		187	BB		219	DB		251	FB	
156	9C	スターキラー (左上)	188	BC	シェル クリーパー (2)	220	DC	第1カーソル	252	FC	OP ER 文 の字 A T O R
157	9D		189	BD		221	DD	第2カーソル	253	FD	
158	9E		190	BE		222	DE	ランプ(1)	254	FE	
159	9F		191	BF		223	DF	ランプ(2)	255	FF	

●キャラクタコード表B (主に、バックグラウンド画用、及びキーボードの文字、記号、
BG GRAPHICに使用できるキャラクタです)
対応する文字の欄に()で書いてある番号は、キャラクターテーブルBのキャラクタに対応します。
[ファミリーページング]
[取扱説明書・P113参照]

コード (10進数)	コード (16進数)	対応する 文字									
0 00	(A0)	32 20		64 40	@	96 60	ア				
1 01	(A1)	33 21	!	65 41	A	97 61	イ				
2 02	(A2)	34 22	-	66 42	B	98 62	ウ				
3 03	(A3)	35 23	#	67 43	C	99 63	エ				
4 04	(A4)	36 24	\$	68 44	D	100 64	オ				
5 05	(A5)	37 25	%	69 45	E	101 65	カ				
6 06	(A6)	38 26	&	70 46	F	102 66	キ				
7 07	(A7)	39 27	-	71 47	G	103 67	ク				
8 08	(B0)	40 28	(72 48	H	104 68	ケ				
9 09	(B1)	41 29)	73 49	I	105 69	コ				
10 0A	(B2)	42 2A	*	74 4A	J	106 6A	サ				
11 0B	(B3)	43 2B	+	75 4B	K	107 6B	シ				
12 0C	(B4)	44 2C	,	76 4C	L	108 6C	ス				
13 0D	(B5)	45 2D	-	77 4D	M	109 6D	セ				
14 0E	(B6)	46 2E	.	78 4E	N	110 6E	ソ				
15 0F	(B7)	47 2F	/	79 4F	O	111 6F	タ				
16 10	(C0)	48 30	0	80 50	P	112 70	チ				
17 11	(C1)	49 31	1	81 51	Q	113 71	ツ				
18 12	(C2)	50 32	2	82 52	R	114 72	テ				
19 13	(C3)	51 33	3	83 53	S	115 73	ト				
20 14	(C4)	52 34	4	84 54	T	116 74	ナ				
21 15	(C5)	53 35	5	85 55	U	117 75	ニ				
22 16	(C6)	54 36	6	86 56	V	118 76	ヌ				
23 17	(C7)	55 37	7	87 57	W	119 77	ネ				
24 18	(D0)	56 38	8	88 58	X	120 78	ノ				
25 19	(D1)	57 39	9	89 59	Y	121 79	ハ				
26 1A	(D2)	58 3A	:	90 5A	Z	122 7A	ヒ				
27 1B	(D3)	59 3B	:	91 5B	「	123 7B	フ				
28 1C	(D4)	60 3C	<	92 5C	¥	124 7C	ヘ				
29 1D	(D5)	61 3D	=	93 5D	」	125 7D	ホ				
30 1E	(D6)	62 3E	>	94 5E	へ	126 7E	マ				
31 1F	(D7)	63 3F	?	95 5F	-	127 7F	ミ				

●バックグラウンド用に直接キャラクタを定義する場合は、このコード表にあるコード(10進数、または16進数)で指定してください。
ただし、コード(16進数00~1F)は、システムで使用しているため直接指定することはできません。
●0~255までの10進と16進の換算表としてもご利用ください。

コード (10進数)	コード (16進数)	対応する 文字									
128	80	ム	160	A0	ゾ	192	C0	(F0)	224	E0	(J0)
129	81	メ	161	A1	ダ	193	C1	(F1)	225	E1	(J1)
130	82	モ	162	A2	ヂ	194	C2	(F2)	226	E2	(J2)
131	83	ヤ	163	A3	ヅ	195	C3	(F3)	227	E3	(J3)
132	84	ユ	164	A4	ヂ	196	C4	(F4)	228	E4	(J4)
133	85	ヨ	165	A5	ド	197	C5	(F5)	229	E5	(J5)
134	86	ラ	166	A6	バ	198	C6	(F6)	230	E6	(J6)
135	87	リ	167	A7	ビ	199	C7	(F7)	231	E7	(J7)
136	88	ル	168	A8	ブ	200	C8	(G0)	232	E8	(K0)
137	89	レ	169	A9	ベ	201	C9	(G1)	233	E9	(K1)
138	8A	ロ	170	AA	ボ	202	CA	(G2)	234	EA	(K2)
139	8B	ワ	171	AB	バ	203	CB	(G3)	235	EB	(K3)
140	8C	ン	172	AC	ビ	204	CC	(G4)	236	EC	(K4)
141	8D	ヲ	173	AD	ブ	205	CD	(G5)星①	237	ED	(K5)
142	8E	ア*	174	AE	ベ	206	CE	(G6)星②	238	EE	(K6)
143	8F	イ*	175	AF	ポ	207	CF	(G7)ホール	239	EF	(K7)
144	90	ウ*	176	B0	口	208	D0	(H0)	240	F0	(L0)
145	91	エ*	177	B1	。	209	D1	(H1)	241	F1	(L1)
146	92	オ*	178	B2	[210	D2	(H2)	242	F2	(L2)
147	93	ヤ*	179	B3]	211	D3	(H3)	243	F3	(L3)
148	94	ユ*	180	B4	◎	212	D4	(H4)	244	F4	(L4)
149	95	ヨ*	181	B5	×	213	D5	(H5)	245	F5	(L5)
150	96	ッ*	182	B6	÷	214	D6	(H6)	246	F6	(L6)
151	97	ガ	183	B7	フ	215	D7	(H7)	247	F7	(L7)
152	98	ギ	184	B8	(E0)	216	D8	(I0)	248	F8	(M0)
153	99	グ	185	B9	(E1)	217	D9	(I1)	249	F9	(M1)
154	9A	ゲ	186	BA	(E2)	218	DA	(I2)	250	FA	(M2)
155	9B	ゴ	187	BB	(E3)	219	DB	(I3)	251	FB	(M3)
156	9C	ザ	188	BC	(E4)	220	DC	(I4)	252	FC	(M4)
157	9D	ジ	189	BD	(E5)	221	DD	(I5)	253	FD	(M5)①
158	9E	ズ	190	BE	(E6)	222	DE	(I6)	254	FE	(M6)②
159	9F	ゼ	191	BF	(E7)	223	DF	(I7)	255	FF	(M7)③

*は小文字です。

ファミリーベーシックV3の仕様

- ソフト：1. BASIC (BG GRAPHICを含む)
2. ハート
3. ペンペン迷路
4. マリオワールド
5. スターキラー

項目		仕様
ROM		プログラム用ROM.....8ビット×32K×1個 キャラクタゼネレータ用ROM.....8ビット×8K
RAM		ワーキング及びメモリ/ソフトアップ用 S-RAM8ビット×4K
表示機能 力	文字表示	28文字×24行 (BASIC時)
	文字構成	8×8ドットマトリックス文字、英文字、数字、カタカナ、英記号、特殊記号、キャラクタ
	表現色数	52色 (白および、黒を含む)
	動画 (スプライト)	16スプライト (ただし、1スプライト 16×16ドット または、8×8ドット) 256×240ドット (BASIC時)
サウンド		2個の方形波音 1個の三角波音 1個の効果音
メモリ/ソフトアップ		BASIC カセット内に、単三乾電池を内蔵する事によりBASIC プログラムや、BGグラフィックも一時的に記憶させておく事ができる。
寸法	カセット (HVC-VT)	110(W)×110(D)×17(H)
重量	カセット (HVC-VT)	95g (電池含まず)
使用条件		使用温度0～40°C、使用湿度35～75%
付属品		ハンドブック

*仕様および外観の一部を予告なく変更する場合がありますのでご了承ください。

BASICの仕様水準

文字の種類	数字・英文字・カナ・記号
数の表現範囲 (整数)	10進 (-32768～+32767) 16進 (&H0000～&FFFF) 文字列 (0～31文字)
変数名の種類	英字で始まる先頭2文字を識別。長さ255桁
行番号の範囲	0～65535
テキスト行の最大桁	255桁
配列の大きさ	2次元まで要素数制限なし (メモリの範囲内)
マルチステートメント	可能、:(コロン)で区切る
サブルーチン、ネスト	制限なし (メモリの範囲内)
FOR NEXTループ	制限なし (メモリの範囲内)
編集機能	スクリーンエディタ
画面モード	BGグラフィック面、スプライト面、バックグラウンド面 バックドロップ面
グラフィック精度	バックグラウンド面 (28文字×24行) スプライト面 (256×240ドット) 1キャラクタ (8×8ドット)
カラー	カラーゼネレータ52色発生
フィルター機能	カラー0～7の着色可能
サウンド機能	音階、テンポ、3重音、音色
マイク音入力機能	音入力の有無検出
コントローラ入力	I II コントローラ (方向入力、トリガー入力可能)
ファイル機能	カセットテープ (1,200ポー)
命令数	基本100
アニメキャラクタ設定	既設16種類より選択

*数の表現範囲である-32768～+32767を越える数値を扱う演算結果については保証されません。

取り扱い上のご注意

1. V3カセットを着脱するときには、必ずファミリーコンピュータ本体の電源スイッチを切ってください。
2. V3カセットを差し込むときは、カセットやファミリーコンピュータ本体に、無理な力が加わらないよう充分注意して、最後までしっかりと差し込んでください。
3. V3カセットを抜き取るときは、イジェクトバーを使用し、静かに抜き取ってください。
4. V3カセットを、熱のこもる場所やほこりの多い場所に置かないでください(故障の原因になります)。
5. V3カセットでのご使用中に異常がありましたときには、すぐに使用を中止して、お買い求めの販売店または最寄の任天堂(株)本社・各支店・営業所に、ご相談ください。
6. V3カセットの分解、改造は絶対におやめください。それ以後のカセットおよび本体への動作異常については、保障いたしませんのでご注意ください。

乾電池取り扱い上のご注意

メモリバックアップのためには、単二乾電池2本が必要です。乾電池は、誤った使い方をしますと、液漏れや破裂する恐れがありますので、次の点にご注意ください。

1. 乾電池の田と日の向きを、V3カセットの表示どおり、正しく入れてください。
2. 新しい乾電池と一度使用した乾電池を、一緒に使わないでください。
3. 種類の違う乾電池を、一緒に使わないでください。同じ形状でも、電圧の異なるものがあります。
4. 乾電池は使えなくなったり、長時間使わないと、V3カセットから取りはずしておいてください。

ご注意

このハンドブックは、NS-HUBASICに基づいて作成されています。

- (1) システムソフトウェアおよび本書の内容は、改良のために予告なく変更することがありますので、バージョンナンバー(NS-HUBASIC V3.0)には、特にご注意されるよう、お願ひいたします。
- (2) 本機のシステムソフトウェアならびに本書の内容を、当社に無断で複製することを禁止します。
- (3) 本機は、非常に複雑な機能および組み合わせを有する製品であり、出荷に際して、本書を含め充分なチェックをして万全を期しておりますが、万一ご使用中ご不審な点、お気付きのことがありましたら、最寄の任天堂(株)本社、各支店・営業所までご連絡ください。
なお、運用した結果生じる影響については、責任を負いかねますので、あらかじめご了承ください。

※製造上の原因で、本品に故障が生じました場合は、修理・交換させて貰ります。

MEMO

ふろく

アルファベット順命令語索引

オート AUTO8,97	ファインド FIND8,106
バックアップ BACKUP11,19,98	フィルター FILTER10,107
ビージーゲット BGGET11,19,99	ゲーム GAME12,17,108
ビージーブット BGPUT11,22,99	インストリング INSTR11,109
ビージーツール BGTOOL11,100	ロード エス LOADS11,23,110
キャンセル CAN9,101	オン エラー ON ERROR GOTO9,111
クリック CLICK11,102	リナンバー RENUM8,112
シーエルエス CLS10,102	リジューム RESUM9,113
クラッシュ CRASH9,103	セーブ エス SAVES11,23,114
デリート DELETE8,103	スクリーン SCREEN10,115
イーアールエル ERL9,104	トレースオン TRON8,116
イーアールアール ERR9,105	トレースオフ TROFF8,116
エラー ERROR9,106	ベクター VCT10,117

MEMO

♪そら

任天堂公式ホームページ

801,8..... FINISH

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

ファミリーベーシック V3 ハンドブック

801,8..... DEFILE

1985年2月19日 2版第1刷発行

801,8..... ERASE

801,8..... 发行人 山内 博

801,8..... 発行所 任天堂株式会社

801,8..... TECO
〒605 京都市東山区福種上高松町60番地
TEL 075(541)6111(代表)

801,8..... ERRC

801,8..... 印刷所 株式会社和多田印刷所

801,8..... VCT

801,8..... 禁無断転載

801,8..... ERROR

801,8..... EAR

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

801,8..... LOADS

801,8..... ON ERROR GOTO a,111

801,8..... RENUM

801,8..... RESUM

801,8..... SAVES

801,8..... ERASE

801,8..... ERRC

801,8..... FCT

801,8..... DELET

801,8..... RESUME

801,8..... CRASH

801,8..... CLS

801,8..... GETPUT

801,8..... BACKUP

801,8..... FILTER

801,8..... GAME

801,8..... INSTR

任天堂株式会社

本 社 〒605 京都市東山区福稻上高松町60番地 TEL (075)541-6111番(代表)

東京支店 〒101 東京都千代田区神田須田町1丁目22 TEL (03) 254-1781番(代表)

大阪支店 〒542 大阪市南区長堀橋筋1丁目32 TEL (06) 271-5514番(代表)

名古屋営業所 〒451 名古屋市西区幅下2丁目18番9号 TEL (052)571-2506番(代表)

札幌営業所 〒060 札幌市中央区北九条西18丁目2番地 TEL (011)621-0513番(代表)

岡山営業所 〒700 岡山市奉還町4丁目4番11号 TEL (0862)52-1821番(代表)